8-2020

Facial Expression Recognition in the Wild Using Convolutional Neural Networks

Amir Hossein Farzaneh
Utah State University

FACIAL EXPRESSION RECOGNITION IN THE WILD USING CONVOLUTIONAL NEURAL NETWORKS

by

Amir Hossein Farzaneh

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Computer Science

Approved:

Xiaojun Qi, Ph.D. Major Professor	Haitao Wang, Ph.D. Committee Member
Curtis Dyreson, Ph.D. Committee Member	Vicki Allan, Ph.D. Committee Member
David Brown, Ph.D. Committee Member	Richard S. Inouye, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

Facial Expression Recognition in the Wild Using Convolutional Neural Networks

by

Amir Hossein Farzaneh, Doctor of Philosophy

Utah State University, 2020

Major Professor: Xiaojun Qi, Ph.D.
Department: Computer Science

Facial Expression Recognition (FER) has demonstrated remarkable progress due to the advancement of deep Convolutional Neural Networks (CNNs). FER's goal as a visual recognition problem is to learn a mapping from the facial embedding space to a set of fixed expression categories using a supervised learning algorithm. Softmax loss as the de facto standard in practice fails to learn discriminative features for efficient learning. Deep Metric Learning (DML) approaches such as center loss, and its variants have been adopted in many FER methods to enhance the discriminative power of learned embeddings. DML fundamentally aims to maximize intra-class similarity and inter-class separation in the embedding space. However, center loss and its variants ignore the two major underlying challenges associated with wild FER datasets, namely, extreme class imbalance and sub-optimal generalization.

Extreme class imbalance leads to a separation bias toward majority classes and leaves minority classes overlapped in the embedding space. To circumvent this issue, we propose a novel Discriminant Distribution-Agnostic loss (DDA loss) to optimize the embedding space for extreme class imbalance scenarios. Specifically, DDA loss enforces inter-class separation of deep features for both majority and minority classes. Any CNN model can be trained with the DDA loss to yield well-separated deep feature clusters in the embedding space.

Equal supervision of all feature elements, where irrelevant elements have the same importance as the relevant elements, leads to sub-optimal generalization. We propose a Deep Attentive Center Loss (DACL) method to adaptively select a subset of significant feature elements for enhanced discrimination. The proposed DACL integrates an attention mechanism to estimate attention weights correlated with feature importance using the intermediate spatial feature maps in CNN as context. The estimated weights accommodate the sparse formulation of center loss to selectively achieve intra-class compactness and inter-class separation for the relevant information in the embedding space.

We conduct experiments on two popular large-scale wild FER datasets (RAF-DB and AffectNet) to show the enhanced discriminative power of our proposed methods compared with several state-of-the-art FER methods.

(99 pages)

PUBLIC ABSTRACT

Facial Expression Recognition in the Wild Using Convolutional Neural Networks

Amir Hossein Farzaneh

Facial Expression Recognition (FER) is the task of predicting a specific facial expression given a facial image. FER has demonstrated remarkable progress due to the advancement of deep learning. Generally, a FER system as a prediction model is built using two sub-modules: 1. Facial image representation model that learns a mapping from the input 2D facial image to a compact feature representation in the embedding space, and 2. A classifier module that maps the learned features to the label space comprising seven labels of *neutral*, *happy*, *sad*, *surprise*, *anger*, *fear*, or *disgust*. Ultimately, the prediction model aims to predict one of the seven aforementioned labels for the given input image. This process is carried out using a supervised learning algorithm where the model minimizes an objective function that measures the error between the prediction and true label by searching for the best mapping function. Our work is inspired by Deep Metric Learning (DML) approaches to learn an efficient embedding space for the classifier module. DML fundamentally aims to achieve maximal separation in the embedding space by creating compact and well-separated clusters with the capability of feature discrimination. However, conventional DML methods ignore the underlying challenges associated with wild FER datasets, where images exhibit large intra-class variation and inter-class similarity.

First, we tackle the extreme class imbalance that leads to a separation bias toward facial expression classes populated with more data (*e.g., happy* and *neutral*) against minority classes (*e.g., disgust* and *fear*). To eliminate this bias, we propose a discriminant objective function to optimize the embedding space to enforce inter-class separation of features for both majority and minority classes.

Second, we design an adaptive mechanism to selectively discriminate features in the embedding space to promote generalization to yield a prediction model that classifies unseen images more accurately. We are inspired by the human visual attention model described as the perception of the most salient visual cues in the observed scene. Accordingly, our attentive mechanism adaptively selects important features to discriminate in the DML's objective function.

We conduct experiments on two popular large-scale wild FER datasets (RAF-DB and AffectNet) to show the enhanced discriminative power of our proposed methods compared with several state-of-the-art FER methods.

To my mother who made me invincible through all the ebbs and flows and to my wife, my muse to ascend the summits.

ACKNOWLEDGMENTS

There are many people I am grateful for contributing to the five thrilling years during my Ph.D. program at Utah State University.

First, I must thank my adviser Xiaojun Qi who, through many emails and meetings, shaped a researcher that I am right now. Her infectious passion and foresight pushed me to achieve my ambitious goals. I am very grateful to Xiaojun for teaching me how to think and how to be independent and strong both in academia and life.

I would like to also thank other members in my Ph.D. committee: Dr. Haitao Wang, Dr. Vicki Allan, Dr. Curtis Dyreson, and Dr. David Brown for their valuable insights and encouragements to increase the horizon of my research.

I have been very fortunate to have the support of my parents from thousands of miles away. My mother, Fatemeh, taught me to be strong in the darkest times and celebrated with me in my happiest moments. My father, Babak, who always acknowledged me for my achievements and encouraged me to learn more. My wife, Leila, who never doubted me and held my back. Her tenacity and professionalism moved me through our years together. My sister, Negar, with whom we laughed a lot to make days even more beautiful.

I have to thank many people who have contributed to my day-to-day life and who have made my experience at Utah State University pleasant: Leyla Ahmady, Mahyar Aboutalebi, Irene Garousi-Nejad, Pedram Jahangiry, Sepideh Raei, little Dara Jahangiry, Mohammad Shekaramiz, Mohammadreza Javanmardi, Elham Hoominfar, Hossein Nasr Esfahani, Manijeh Nouraei and her family, Myrna Redd, Gayle Parkinson, and Spencer Parkinson.

Amir Hossein Farzaneh

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Seeking faces and analyzing facial attributes is an active field of research in deep learning. Facial expressions as a non-verbal channel to convey emotions and intentions are one of the most universally powerful tools in communication. Researchers in computer vision motivated by a wide range of applications have become increasingly interested in designing and implementing automatic systems that recognize facial expressions. Such systems are very needed to substantiate and expedite the analysis of human behavior in a digital environment. Facial Expression Recognition (FER) is an essential tool to detect emotions and has been widely used in many aspects of modern society, such as healthcare, autonomous driving, human-computer interaction, and education. As shown in Figure 1.1, a FER system predicts an expression using the facial features.

Emotions are represented in different ways, such as Facial Action Coding System (FACS) [1], dimensional affect (*e.g.,* valence and arousal) [2, 3], and categorical facial expressions (*e.g., neutral*, *happy*, *sad*, *surprise*, *anger*, *fear*, or *disgust*). Annotating visual channels (video or images) with FACS or dimensional affect is a very rigorous and tedious

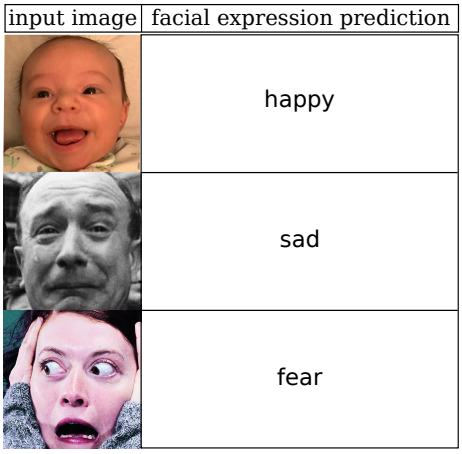


Fig. 1.1: Example FER system predictions

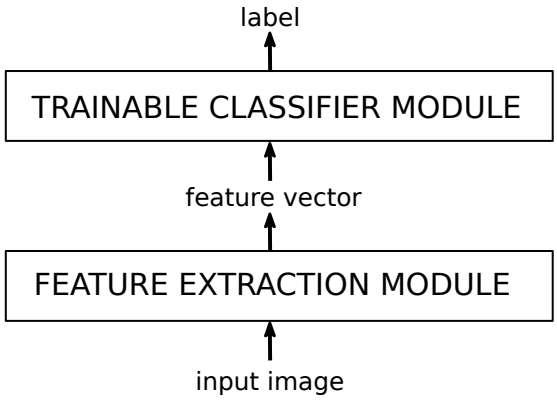


Fig. 1.2: A high level representation of a FER system where the input image is a 2D facial image and the output is an expression label.

task and requires specially trained professionals. Consequently, categorical FER models are popular in the field due to their simple interpretation, relative ease of data collection, and wide applicability. Ekman and Friesen [4] argue that the implied meaning of an expression can be labeled with six basic expression categories that are universally the same across different cultures: *happy*, *sad*, *surprise*, *anger*, *fear*, and *disgust*. *Neutral* expression is subsequently added to this set of prototypical expressions to comprise seven expressions commonly explored in FER research. A FER system that assigns one of these expressions to a facial image uses a categorical model. In this dissertation, we focus on categorical FER systems.

Given a 2D facial image, FER as a visual recognition task involves two major subtasks: 1. feature extraction to represent facial appearances, and 2. feature classification to recognize facial expressions by mapping features to a fixed set of expression labels. Figure 1.2 shows a high level diagram of a FER system. Instead of directly feeding an input image with tens of thousands of pixels to a classifier, the input facial image is fed to a feature extraction module yielding a d-dimensional feature vector. The resulting feature vector can be easily matched and compared to other feature vectors. Furthermore, the feature extraction module maintains most of the important information specific to the task and throws away the input noise. Finally, a trainable classifier module assigns a label to the input image by classifying the feature vector using a machine learning algorithm.

FER systems are trained in an end-to-end manner, meaning that the entire sequence of sub-tasks from the input (facial image) to the output (expression label) shares the same objective of correctly classifying the input.

For decades, the feature extraction module was carefully engineered and required task-specific expertise to transform raw image data into an interpretable feature representation. Conventional FER methods use hand-crafted features, such as Local Binary Patterns (LBP) [5], LBP on three orthogonal planes (LBP-TOP) [6], non-negative matrix factorization (NMF) [7], Histogram of Oriented Gradients (HOG) [8], and sparse learning [9]. Such features focus on low-level spatial attributes and are optimized for lab-controlled FER datasets. Consequently, manually designed features are insufficient for unconstrained natural scenarios with large facial variations and complexities.

FER2013 [10] and Emotion Recognition in the Wild (EmotiW) [11] are pioneering challenges that started to collect large-scale FER training data to promote the performance of FER systems in real-world scenarios. Due to the in-the-wild attribute, large-scale facial expression datasets acquired in an unconstrained environment inherently populate expression categories with significant variations in pose, gender, age, demography, image quality, and illumination. This new wave of FER research requires more advanced feature extractors to capture the variations and complexities efficiently.

Advances in deep learning over recent decades have led to a growing interest in the development of deep learning-based approaches to FER. With modern hardware and storage abundance in recent years, there has been an explosion of research in computer vision tasks using Deep Neural Networks (DNNs). Collecting data is easier than before, and more massive datasets are available for researchers. The computer science community is now able to write complex algorithms to look at the data, analyze the data, and identify patterns. This achievement is possible with powerful Graphical Processing Units (GPUs) with thousands of parallelized computing cores.

DNN methods have dramatically improved the state-of-the-art visual recognition [12, 13]. Mainly, the emergence of Convolutional Neural Networks (CNNs) [14] as a dominant

deep learning technique has offered an advanced tool for researchers to overcome the complications with real-world image data [15]. At their core, CNNs automatically learn complex image features in a hierarchical manner to yield high-level representations that encode the abstract semantics of the data. A trainable classifier unit then separates the resulting deep feature vectors into classes using a designed objective function. State-of-the-art deep CNN models require a massive corpus of labeled data to learn powerful image features [16]. Similarly, FER benefits greatly from training deep CNNs on large-scale FER datasets acquired in real-world scenarios [10, 11].

1.1 Deep Learning Background

Deep learning is a specific type of machine learning. Many concepts that are practiced in deep learning are derived from the original concepts that are practiced in classical machine learning. Specifically, the task of Facial Expression Recognition (FER) is a supervised learning problem where a label or target is assigned to an input based on supervised model trained on a FER training dataset.

Our work is inspired by modern deep learning methods in FER. In this section, we provide the relevant mathematical background of deep learning used in our work. We formally review the necessary concepts in supervised learning, loss function to measure the prediction error, optimization, and Convolutional Neural Networks (CNNs) to build an image recognition system.

1.1.1 Supervised Learning

Generally, solving a computer vision problem using machine learning is an optimization problem that searches for the following mapping function f with associated parameters θ:

$$f_\theta : X \to Y \tag{1.1}$$

where X is the input space, and Y is the output space. For a visual recognition task, which is essentially a multi-class image classification problem, the input space is a 2D image, and the

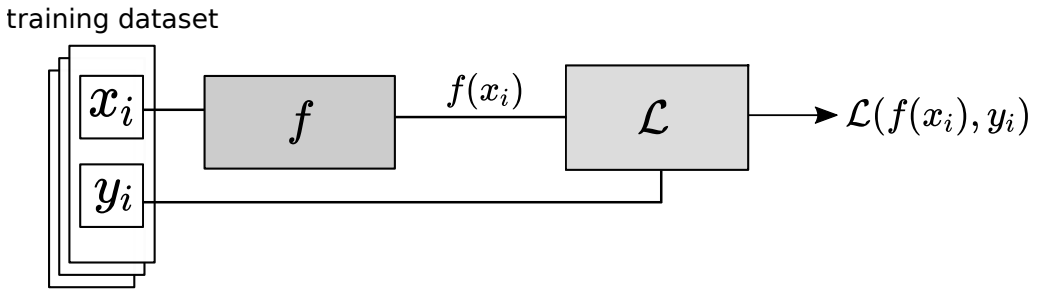


Fig. 1.3: Flow of data in a typical supervised learning problem. The input is a training dataset of m samples in pairs of (x_i, y_i). The function f maps the input instances x_i to a predicted label $f(x_i) = \hat{y}_i$. The loss function $\mathcal{L}$ measures the discrepancy between the predicted label and the true label y_i.

output space is a label or target assigned to the input from a fixed set of categories specific to the problem. In an image classification problem, manually specifying f is impractical. On the other hand, given an outcome is provided for all input samples, *supervised learning* offers useful techniques to approximate f. Intuitively, supervised learning aims to discover a pattern that is shared by all the input data samples belonging to a class by experiencing the training dataset. The discovered pattern is used for prediction on new data samples.

Formally, the goal of supervised learning is to approximate the mapping function f to create a prediction model based on a training dataset, where each input is associated with a label. Assuming a training dataset of n sample pairs:

$$\{(x_1, y_1), ...(x_n, y_n)\} \tag{1.2}$$

where $(x_i, y_i) \in X \times Y$ and $i \in \{1, \cdots, n\}$, the supervised learning's objective is to find f^* in the space of functions $\mathcal{F}$ that minimizes a loss function $\mathcal{L}(\hat{y}_i, y_i)$ over all samples in the training dataset:

$$f_\theta^* \approx \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i, \theta), y_i) \tag{1.3}$$

The loss function $\mathcal{L}$ measures a disagreement between a predicted label $\hat{y}_i = f(x_i, \theta)$ and a true label y_i. The diagram for a typical supervised learning problem is presented in Figure 1.3.

The performance of a prediction model is usually measured by *accuracy*, which is the proportion of correctly classified data samples to all data samples. In practice, we are interested in the performance of a prediction model on previously unobserved data or its *generalization* ability on a testing dataset that is separate from the training dataset. The ability of the prediction model to provide a good outcome for a new unobserved data is called *generalization*. The constructed prediction model f_θ^* captures the hidden information in the training dataset and ideally is able to generalize well on the testing dataset. The common practice is to provide a testing/validation dataset as unobserved data that accompanies the training dataset to assess the validity of the model. If a testing/validation dataset is not available, the training dataset is split into a training dataset and a testing/validation dataset.

For a supervised learning algorithm to be applicable in real-world settings, the training dataset should be rich in sample quantity, variability, and integrity. Furthermore, the prediction model must be capable of generalizing to unobserved real-world samples. This means that the algorithm should be able to discover complex patterns from a large number of samples belonging to a class that exhibit diverse visual variations, and possibly noise. Classical machine learning models such as neural networks and support vector machines (SVMs) are insufficient for real-world tasks. However, deep learning offers solutions that overcome the limitations of classical machine learning techniques.

Deep learning in the context of supervised learning is commonly approached with DNNs. DNNs can discover complex patterns from the input data and transform the input space into an efficient representation for classification. Current hardware capabilities can handle multiple stacked layers with millions of parameters to map latent information from one layer to another and simultaneously learn non-linear relationships between layers. Subsequently, a feature vector is learned automatically without human intervention.

1.1.2 Loss Function

Choosing an appropriate loss function specific to the problem that is being solved is a critical step. In this dissertation, we have explored different loss functions to achieve an

efficient mapping from the input space to the output space to tackle the challenges with real-world FER. In this section, we briefly review the necessary mathematical background for designing a loss function.

In a typical multi-class image classification problem, the mapping function $f_\theta : X \to Y$ maps the input space to a probability distribution $\hat{y}_i \in Y$ over all classes. The true labels in the dataset are also represented as a probability distribution y_i. Given the predicted label $\hat{y}_i$ for an arbitrary sample from the dataset and its corresponding true label y_i, the objective function optimizes the parameters θ such that two probability distributions $\hat{y}_i$ and y_i are similar.

One way to measure the dissimilarity between two probability distributions is known as **Kullback-Leibler divergence** (**KL divergence**). KL divergence, also known as relative entropy is defined as follows:

$$\mathbb{KL}(y_i \| \hat{y}_i) = \sum_{k=1}^{K} y_{ik} \log \frac{y_{ik}}{\hat{y}_{ik}} \tag{1.4}$$

where $y_{ik} = 1$ if x_i belongs to the k-th class and 0 otherwise, $\hat{y}_{ik}$ represents the predicted probability of an input sample to be in a class k, and K is the number of classes. Equation 1.4 can be re-written as:

$$\mathbb{KL}(y_i \| \hat{y}_i) = \sum_{k} y_{ik} \log y_{ik} - \sum_{k} y_{ik} \log \hat{y}_{ik} = -\mathbb{H}(y_i) + \mathbb{H}(y_i, \hat{y}_i) \tag{1.5}$$

where $\mathbb{H}(y_i, \hat{y}_i)$, known as cross-entropy or negative log-likelihood, measures the dissimilarity between the predicted label and the true label:

$$\mathbb{H}(y_i, \hat{y}_i) = -\sum_{k} y_{ik} \log \hat{y}_{ik} \tag{1.6}$$

In the supervised learning paradigm, we minimize $\mathbb{H}(y_i, \hat{y}_i)$ to minimize the dissimilarity between the true label and the prediction of the model f^*. Equation 1.6 is the foundation of many supervised learning methods. Compared to the standard Mean Squared Error

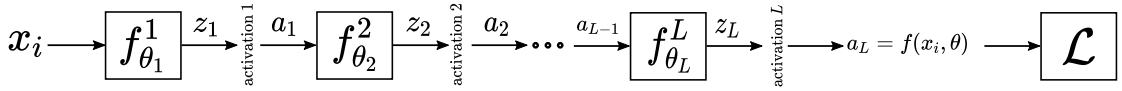


Fig. 1.4: The computational graph for the forward-propagation. The input sample x_i from the dataset is forward-propagated through a series of mapping functions $f_\theta = \{f^1_{\theta_1}, f^2_{\theta_2}, ..., f^L_{\theta_L}\}$ to generate the predicted output of the network $\hat{y}_i = f(x_i, \theta) = a_L$. Finally, the loss function measures the discrepancy between the final output and the true label y_i.

(MSE), cross-entropy offers three advantages: 1. It does not emphasize the incorrect class predictions, 2. It is easily optimized with standard optimization algorithms (*e.g.,* Gradient Descent algorithms), and 3. It is not affected by the curse of dimensionality.

In deep learning, the loss function L is defined as the average cross-entropy for a batch of m samples:

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{m} \mathcal{L}(f(x_i), y_i) \\ &= -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_{ik} \log \hat{y}_{ik} \end{aligned} \tag{1.7}$$

1.1.3 Optimization

Training Deep Neural Networks (DNNs) is achieved in three stages: 1. Forward-propagation, 2. Back-propagation, and 3. Optimization with a standard Gradient Descent algorithm.

Forward-propagation. In forward-propagation stage, the i-th input sample is forward-propagated through a chained sequence of mapping functions $f_\theta = \{f^1_{\theta_1}, f^2_{\theta_2}, ..., f^L_{\theta_L}\}$, where θ_l are the set of parameters for layer l, and L is the total number of layers in the network. Each layer performs a linear transformation on its input x_l and the layer's output z_l is activated with a non-linear function $a_l = \text{activation}_l(z_l)$. The input to each layer is the activated output from the previous layer. Finally, the loss function measures the discrepancy between the final output $\hat{y}_i = f(x_i, \theta) = a_L$ and the true label y_i. The computational graph for a forward-propagation process is depicted in Figure 1.4.

Back-propagation. Rumelhart *et al.* [17] pioneered the application of back-propagation to neural networks as a faster alternative learning method. In the back-propagation step, the gradient (partial derivatives) of the loss function $\mathcal{L}$ with respect to the network parameters θ are calculated using the chain rule. In Figure 1.4 the gradients are propagated in the opposite direction of the data flow in forward-propagation.

Gradient Descent Optimization. Gradient Descent is the widely used algorithm to optimize deep neural networks. For a large dataset, the mini-batch Stochastic Gradient Descent (SGD) is used in practice. Mini-batch SGD iteratively updates the network parameters in the opposite direction of their gradients (calculated in back-propagation) with small increments. Algorithm 1 summarizes an iteration of the SGD update.

Algorithm 1: The standard mini-batch Stochastic Gradient Descent algorithm for one iteration

Input:
The learning rate hyper-parameter μ;
Initial parameters θ.
Output: Updated parameters θ^*.

while *a stopping criterion is not met* **do**
  Sample a mini-batch of size m from the training dataset.
  Compute the gradient of loss function with respect to the parameters θ:
  $\hat{\boldsymbol{g}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta} \mathcal{L}\left(f\left(x_i, \theta\right), y_i\right)$
  Update the network parameters: $\theta^* = \theta - \mu \hat{\boldsymbol{g}}$.

Choosing an optimized learning hyper-parameter, usually selected with trial and error, is a crucial procedure while training DNNs. Furthermore, the network parameters θ are commonly initialized with the *He* method [18], which preserves the variance of activation between DNN layers. The *He* method initializes the network parameters using a normal distribution $\mathcal{N}(0, \sigma^2)$, where the standard deviation is a function of the size of the previous layer.

In practice, SGD with momentum [19] and weight decay [20] is used to ensure faster convergence and better generalization, respectively.

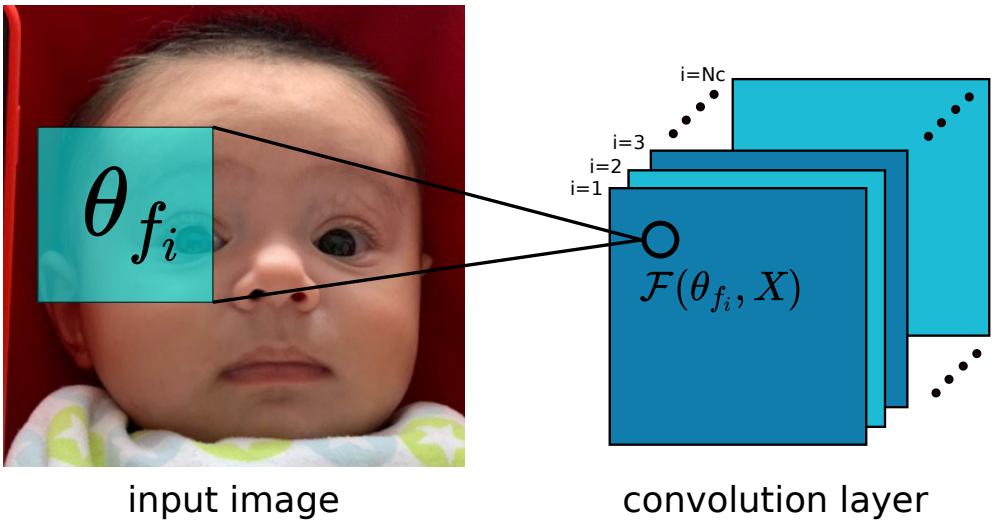


Fig. 1.5: Convolution operation on an input image. The i-th filter with parameters θ_{f_i} is convolved with image pixel values X in a local neighborhood and the response is saved as the filter response $\mathcal{F}(\theta_{f_i}, X)$, where $\mathcal{F}$ denotes the convolution operation.

1.1.4 Convolutional Neural Networks

CNN as a dominant tool in computer vision tasks, is a specialized class of neural networks developed for 2D inputs. Since 1998 [14], CNNs have been applied to a broad variety of computer vision tasks with great success [16]. CNNs offer an advanced tool for researchers to overcome the complications with real-world image data [12].

At the heart of a CNN lies the convolution filters that are convolved locally with the input, and the resulting response represents local features. Fig 1.5 shows an example convolution operation on a patch of the input image. The i-th filter f_i where $i = \{1, 2, ..., N_c\}$ with its corresponding parameters θ_{f_i} is convolved with the image patch pixel values X, and the response is saved as the filter response $\mathcal{F}(\theta_{f_i}, X)$. The same filter is moved along local regions in the input image to create a 2D filter output. Different filters are similarly applied to the input image to construct a convolutional layer with N_c channels.

The convolution outputs are passed through an activation function such as Rectified Linear Unit (ReLU) [21] to extract the hidden non-linearity. Before passing the activated output to the next layer, a max-pooling layer down-samples and extracts significant activations for the next layer.

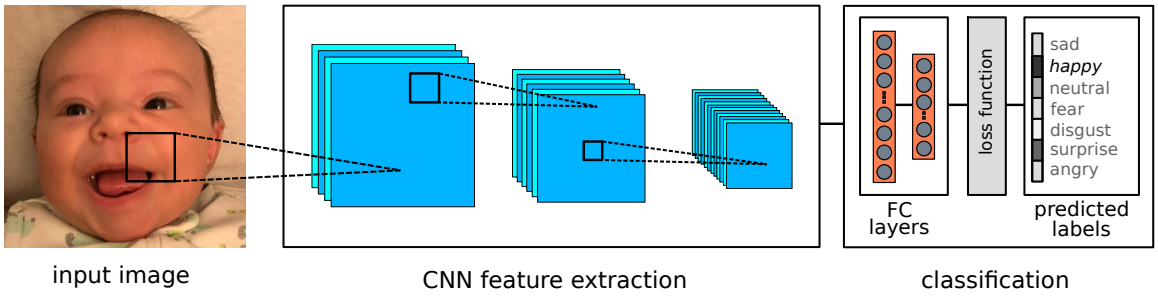


Fig. 1.6: A typical CNN for FER where the input is a 2D facial image and the output is a probability distribution over all classes (seven basic expressions). The input image is fed to a convolutional network, which is constructed with stacked convolutional layers in sequence. The last convolutional layer yields deep features that are pooled with fully-connected layers. A loss function then classifies the resulting deep feature and makes a prediction based on a fixed set of categories.

The advantages of convolution operations in CNNs are two-fold: 1. Parameter sharing, and 2. The sparsity of connections. A convolution filter or a feature detector is applied to different regions of the image *i.e.,* the parameters of the i-th filter is shared for all parts of the image. Furthermore, filter output values depend only on a small number of inputs values which result in sparse connections between the input and the output. Consequently, the designed CNN comprises of fewer parameters compared to its equivalent DNN.

Multiple layers of convolution layers are stacked in sequence to construct a deep CNN. At their core, deep CNNs automatically learn complex image features in a hierarchical manner to yield high-level representations (deep features) that encode the abstract semantics of the data. A trainable linear unit (fully-connected layers) then classifies the resulting deep feature vectors using a specific loss function. For deep CNNs to generalize well on real-world data, a massive corpus of labeled data is required to learn powerful visual features. A typical CNN architecture used in the task of FER is depicted in Figure 1.6. The input facial image is fed to the convolutional network that is constructed by stacking convolutional layers in sequence. The convolutional network learns facial features at many different levels of abstraction from small edges to very complex features such as the nose, eye, and mouth in deeper layers. Deep features are pooled with fully-connected linear units from the last convolutional layer. A loss function then classifies the resulting deep feature and makes a

prediction based on a fixed set of categories.

Many deep learning frameworks are currently developed with easy interfaces for the research community and industry. PyTorch [22], our framework of choice, offers powerful Application Program Interfaces (APIs) that are easily extensible. The customizable modules in PyTorch enable us to have full control of every element of training DNNs. The imperative pythonic programming style in PyTorch makes debugging easier.

Perhaps, the most attractive feature of PyTorch is its automatic differentiation (AutoGrad) system. With AutoGrad, the gradients are subsequently calculated as the data flows in the network computational graph. This behavior leads to faster back-propagation while training networks.

1.2 Wild Facial Expression Recognition Datasets

Facial Expression Recognition (FER) using Deep Neural Networks (DNNs) require a sufficient amount of annotated data that exhibits variations in age, demography, gender, and image quality. However, collecting and annotating data on a large scale is a tedious task and should be supervised by professionals. In this section, we review the characteristics of publicly available FER datasets.

Facial Expression Recognition (FER) datasets are generally divided into two categories: 1. Lab-controlled small datasets, and 2. Natural large-scale datasets.

Lab-controlled FER datasets. Datasets such as the Extended Cohn-Kanade dataset (CK+) [23], MMI [24], the Japanese Female Facial Expression (JAFFE) [25], and Oulu-CASIA [26] are captured in constrained environments where subjects are explicitly asked to portray basic facial expressions in a uniform way. Some examples from a lab-controlled FER dataset is shown in Figure 1.7. Although this type of data collection results in a high-quality and clean dataset, the facial expressions do not represent natural real-world scenarios. Furthermore, the quantity of data is not sufficient for a DNN to generalize well.

Natural large-scale FER datasets. For a FER system to perform well in a real-world setting, datasets with a large number of facial images and variation is required. FER2013 [10], Real-World Affective database (RAF-DB) [27], and AffectNet [28] are such

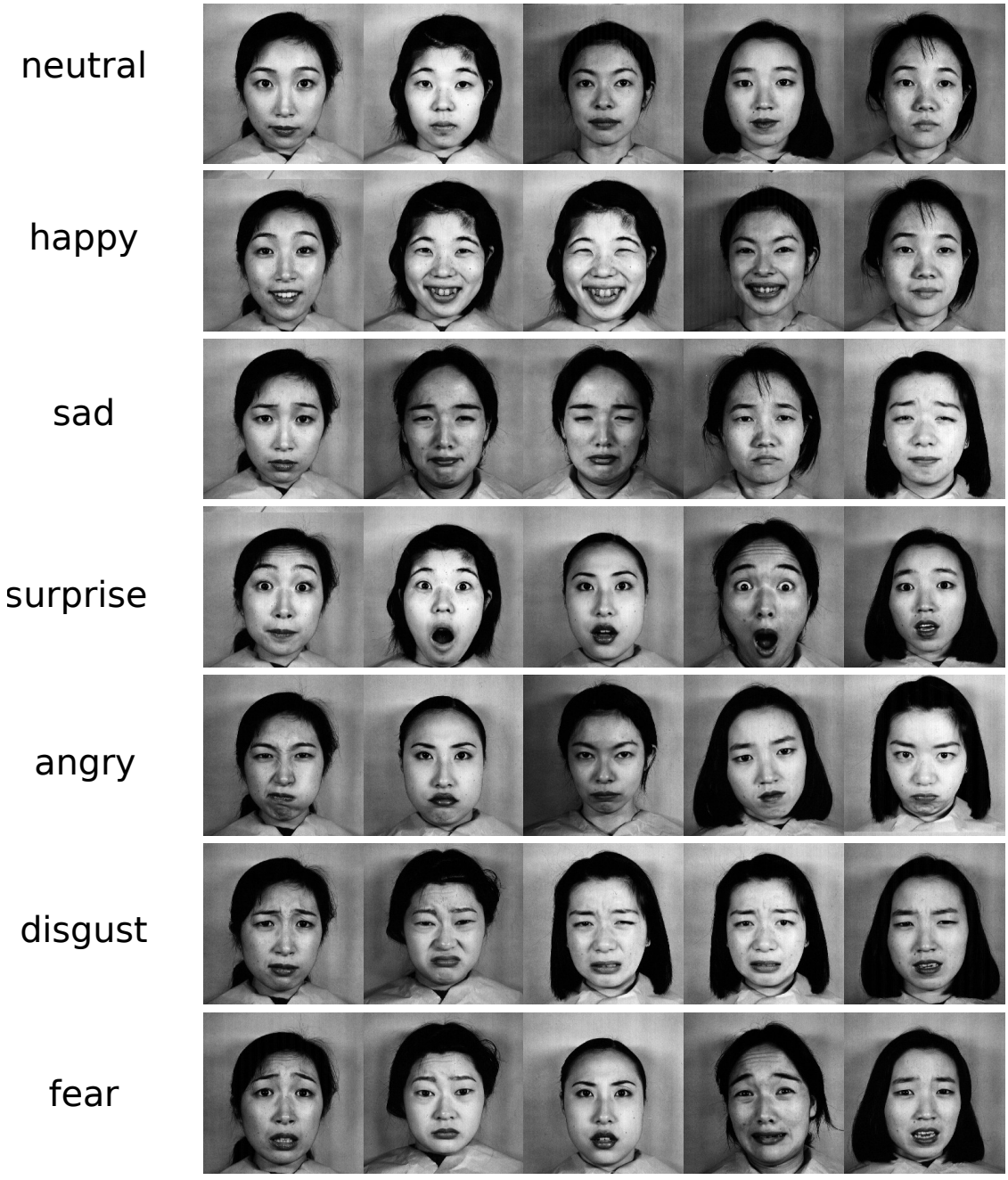


Fig. 1.7: Example images from the Japanese Female Facial Expression (JAFFE) dataset, where each row has a few example images from a facial expression category. Empirically, the subject has been asked to portray the required facial expressions to create a uniform dataset.

datasets. Typically, these datasets are collected with crowd-sourcing techniques such as crawling the web with keywords (facial expressions) and are manually cleaned to remove irrelevant or non-face images. However, these large-scale FER datasets contain label noise *i.e.,* few images in each category are mislabeled. On the other hand, it has been proven that some amount of label noise can help DNN generalization [29].

Raw large scale FER datasets that are acquired in an unconstrained environment such as the internet offer a large diversity across pose, gender, age, demography, image quality, and illumination. Hence, they exhibit an in-the-wild attribute and are called wild FER datasets. Some examples from a wild FER dataset are shown in Figure 1.8. In this dissertation, we are interested in the application of FER in real-world scenarios. Consequently, we design and develop our recognition models in the wild setting trained on wild FER datasets.

1.3 Challenges

Recognizing the basic universal expressions identified by Ekman and Friesen [4] can be challenging even for humans. While a person can easily distinguish between an image of a horse and an image of a ship from the canonical CIFAR10 image classification dataset [30], distinguishing between a person expressing *neutral* and another person expressing *happy* can be a matter of debate. Hence, expression categories often exhibit large intra-class variations and inter-class similarities. Although described as universal, expressions can be perceived differently from one person to another. For example, one might confuse *surprise* with *fear* and vice versa.

The imbalanced distribution of data among classes is another learning obstacle associated with Facial Expression Recognition (FER) datasets. An intrinsic imbalance, a prevalent issue in many real-world data [31], is an inevitable side effect of facial expression categorization. For example, categories such as *fear* and *disgust* are minority classes due to lack of representative data. Other expressions *i.e., neutral*, *happy*, *sad*, *surprise*, and *anger* as majority classes are represented with fair amount of data. Wild FER datasets exhibit an amplified imbalance issue called an extreme class imbalance. For example, the *happy* class

Fig. 1.8: Example images from the AffectNet dataset, where each row shows a few example images from a facial expression category. Empirically, the inter-class and intra-class variations in pose, gender, age, demography, image quality, and illumination yields a diverse dataset that is sufficient for real-world FER applications.

in AffectNet [28] has 134,415 data samples (47% of the whole dataset) while the *disgust* class is represented with 3,803 data samples (1% of the whole dataset). Learning algorithms will develop a bias toward the majority class and will perform poorly on minority classes.

A FER model using a deep Convolutional Neural Network (CNN) transforms millions of pixels of varying brightness into high-level feature vector to represent an emotion such as *happy*. The large intra-class variation and inter-class similarity, extreme class imbalance, and data complexity pose a challenge for learning algorithms to yield a unique representation (pattern) for the images belonging to a class.

1.4 Outline of Contributions

Constructing a prediction model that efficiently maps the input space to the output space requires new algorithms. In this dissertation, we develop models and optimization techniques to tackle the problem of Facial Expression Recognition (FER) in the wild setting. For instance, we aim to design a prediction model that can transform the input image to a feature vector in the embedding space that is distinguished with feature vectors of other classes. We explore deep metric learning approaches and propose a novel loss function that learns representation clusters that are distinctly segregated in the embedding space.

1.5 Structure of the Dissertation

In this dissertation, we develop models that predict a facial expression from facial images in real-world scenarios. Our work is inspired by recent research in the field that aspired to train prediction models with wild Facial Expression Recognition (FER) datasets. In **Chapter 2**, we review the recent related works. **Chapter 3** introduces a new Deep Metric Learning (DML) approach to tackle wild FER under class imbalance scenarios. **Chapter 4** introduces a hybrid of DML and attention mechanism to improve FER model generalization. Finally, we conclude and present a summary of our work in **Chapter 5**.

CHAPTER 2

RELATED WORKS

The work in this dissertation is inspired by deep metric learning approaches where the embedding space is constrained using similarity metrics to enhance the discrimination power of deep feature representations. Most of the metric learning methods have been developed for the Face Recognition (FR) task. However, both FR and Facial Expression Recognition (FER) tasks operate on facial representations. Consequently, the same metric learning methods might benefit FER. In fact, a significant amount of research has been conducted to boost FER performance. In this section, we review previous work in the field from two perspectives: 1. FER using discriminative loss functions, and 2. FER in the wild.

2.1 Facial Expression Recognition Using Discriminative Loss Functions

A widely used solution to improve Facial Expression Recognition (FER) is Deep Metric Learning (DML) which tackles large intra-class variation and large inter-class similarity. DML aims at mapping the raw input to a regularized embedding space where deep features exhibit enhanced discrimination compared to the deep features mapped by the conventional methods such as softmax loss. Intuitively, DML approaches optimize a similarity metric between samples in a mini-batch so that the learned deep features are efficient for classification. The outcome of such methods is well-clustered deep features in the embedding space.

Contrastive loss [32] and triplet loss [33] are two fundamental methods in metric learning. Commonly, a classification problem that is solved by contrastive loss operates on input pair samples in a mini-batch. Contrastive loss attempts to decrease the distance between positive examples with similar targets smaller than a fixed threshold. On the other hand, the distance between negative pairs with different targets is increased larger than a fixed threshold.

Triplet loss operates on a triplet of an anchor, a positive sample, and a negative sample. Theoretically, the positive sample has the same target as the anchor, while the negative sample is associated with a different target. Triplet loss attempts to decrease the positive-anchor distance smaller than the negative-anchor distance constrained by a margin of m. Compared to contrastive loss, triplet loss integrates fewer constraints in the embedding space allowing for variance in inter-class dissimilarities. In this section, we review recent FER methods that take advantage of deep metric learning.

Large intra-class variation is partly the outcome of different identities existing in the same expression category. Therefore, the performance of a FER model might be degraded due to the lack of discrimination between identities. To overcome this issue, Meng *et al.* [34] develop an Identity-Aware Convolutional Neural Network (IACNN) that simultaneously utilizes both expression-related and identity-related deep features. During the training process, an input image pair is forward-propagated through two identical CNNs with shared parameters to jointly calculate the expression-related and identity-related deep features for both images in the input. The softmax loss function is applied on top of the expression-related deep features to calculate the classification error and optimize the network for learning expression-related deep features. Concurrently, an expression sensitive contrastive loss takes the two estimated expression-related deep features from both images and minimizes the Euclidean distance for those that have similar expressions and maximizes the Euclidean distance otherwise.

To tune the model for an identity-aware property, a contrastive loss function is applied to the extracted identity-related deep features to pull those with similar identity toward each other and pull those with different identities away from each other. Finally, the expression-related feature and the identity-related feature are concatenated to represent two images in the input pair. Softmax loss is applied on top of the final representation to measure the final classification error for two images. At test time, one input image is fed through one of the CNNs, and the final prediction is made with the softmax loss function. IACNN is evaluated on FER2013 [10], CK+, SFEW [35], and MMI datasets.

Similarly, Guo *et al.* [36] introduce Deep Neural Networks with Relativity Learning (DNNRL) based on the triplet loss to pull the samples with the same expression towards each other and push those with different expression away from each other in the embedding space. During training, triplets are mined from the dataset including a positive sample, a negative sample, and an anchor. The positive sample shares the same expression with the anchor, and the negative sample has a different expression than the anchor and the positive sample. The triplet loss optimizes the triplet's representation distances in the embedding space so that the positive sample is closer to the anchor than the negative sample with a pre-determined gap of τ. To account for the difficult samples, DNNRL assigns a larger weight for difficult samples based on the output of the network. DNNRL is evaluated on FER2013 and SFEW.

Triplet loss performance is very sensitive to the selection of the anchor example. Liu *et al.* [37] propose (N+M)-tuplet clusters loss function adapted from (N+1)-tuplet loss [38] and Coupled Clusters Loss (CCL) [39] to address the difficulty of anchor selection in triplet loss. Inputs are mined as a set of N positive samples and a set of M negative samples. During training, (N+M)-tuplet clusters loss function forces the samples in the negative set to move away from the center of positive samples and simultaneously clusters the positive samples around their corresponding center to achieve compactness. (N+M)-tuplet clusters loss is evaluated on CK+, MMI, and SFEW.

Although mining based discriminative loss functions such as the contrastive loss and triplet loss improve FER, searching for the input pairs or triplets can be challenging. On the other hand, the training algorithms converge slower as the number of pairs or triplets grow in a quadratic and cubic way, respectively, as the mini-batch size increases. Instead of sample mining, center loss [40] introduces an additional objective function coupled with softmax loss to enhance the discriminative power of deep features. Specifically, center loss creates a compact representation of deep features by minimizing the euclidean distance between the learned deep features to their corresponding class centers.

Locality-Preserving loss (LP-loss) [41], inspired by center loss, enforces intra-class compactness by locally clustering deep features using the k-nearest neighbor algorithm. The proposed Deep Locality-Preserving CNN (DLP-CNN) preserves the locality of each sample's deep representation in the embedding space. During training, the k-nearest neighbors for each sample are searched based on the Euclidean distance. Then, the distance between the sample and the mean of k-nearest neighbors is minimized. LP-loss is evaluated on CK+, SFEW, MMI, and RAF-DB.

Center loss implicitly yields inter-class separation by explicitly achieving intra-class compactness. Therefore, the feature clusters might still be overlapped in the embedding space. To circumvent this issue, Cai *et al.* [42] improve on center loss by adding an extra objective function to achieve intra-class compactness and inter-class separation simultaneously. The modified center loss called Island loss is defined as the summation of the center loss and the pairwise cosine distance between the class centers in the embedding space. During training, the cosine distance is maximized to separate the centers learned by center loss angularly. Island loss is evaluated on CK+, MMI, and Oulu-CASIA.

Similarly, Li *et al.* [43] introduces separate loss to address large intra-class variation and inter-class similarity. Separate loss is a cosine version of center loss and island loss. It consists of two parts: 1. Intra-class loss, and 2. Inter-class loss. The intra-class loss is the normalized cosine similarity between a sample's deep feature representation, and the inter-class loss is the normalized cosine similarity between the centers in the embedding space. During training, the intra-class loss is minimized, and the inter-class loss is maximized. Since both loss functions are based on the normalized cosine similarity metric, they are considered to be commensurate. Hence, a parameter to balance the two loss functions are not required when they are added together. However, when coupled with softmax loss, a hyper-parameter λ is still required to control the contribution of separate loss to the total loss. Separate loss is evaluated on RAF-DB and AffectNet.

Li *et al.* [44] propose a multi-scale CNN with an attention mechanism to learn the importance of different convolutional receptive fields in the network. Additionally, the

softmax loss function is jointly supervised with a regularized version of the center loss to incorporate a distance margin. The multi-scale CNN is composed of multi-scale convolutional filters interjected between layers to combine global and local information from the input. The regularized center loss restricts the distance between the deep feature and its corresponding class center by a margin of α_1. Simultaneously, the distance between the class centers in the embedding space is restricted by an additional margin of α_2. In other words, the intra-class distances are constrained to be less than the margin α_1, and the inter-class distances between the class centers are constrained to be less than the margin α_2. The authors evaluate their method on CK+ and Oulu-CASIA.

2.2 Facial Expression Recognition in the Wild

Most of the previous Facial Expression Recognition (FER) in the wild research can be categorized in two domains: 1. Collecting large datasets and providing benchmarks for the research community, 2. Developing state-of-the-art methods to tackle the wild FER challenges. In this section, we first introduce the methods of collecting two large popular wild FER datasets and then we review recent state-of-the-art methods that tackle the associated challenges with wild FER datasets using Deep Neural Networks (DNNs).

2.2.1 Wild Facial Expression Recognition Dataset Collection

Mollahosseini *et al.* [28] collect the largest wild FER dataset called Affect from Internet (AffectNet). AffectNet is annotated with categorical expressions (*e.g.,* six basic expressions, *neutral*, and *contempt*) and dimensional affect (valence and arousal). The images are acquired from the internet by searching for expression keywords in different languages. The face bounding boxes are extracted using the OpenCV face recognition toolkit. Moreover, 66 face landmarks are provided using local binary features [45] for face alignment purposes. 450,000 images out of one million facial images acquired from the web are manually annotated by 12 annotators. The final training set includes 287,651 images and the validation set includes 4,000 images. The testing set is not released by the authors as of the time of writing this dissertation.

To establish a baseline for the dataset, authors train AlexNet [12] on AffectNet using three methods to circumvent the class imbalance: 1. Down-sampling, 2. Up-sampling, and 3. Weighted-Loss. In the down-sampling method, the data from the classes with more samples are sampled less in the mini-batch compared to other classes. Similarly, in the up-sampling method, the data from the minority class are sampled more in a mini-batch compared to other classes. In the weighted-loss approach, the cross-entropy is weighted based on the number of samples in each class. In other words, the cross-entropy measure is penalized relatively more for the minority class in comparison to the samples in the majority class. Intuitively, the loss function puts more focus on the samples from the minority class. The weighted-loss approach yields an accuracy of 58% on the validation set.

Real-world Affective Face Database (RAF-DB) [27] is the second widely used wild FER dataset. Similar to AffectNet, RAF-DB's facial images are downloaded from the internet using six basic expressions and *neutral* as keywords. 315 annotators were hired and trained to annotate the downloaded images. To address the annotation disagreement between the voters for an image, the authors developed an Expectation-Maximization (EM) algorithm [46] to filter out noisy labels. As a result, RAF-DB exhibits more reliable labels compared to AffectNet. There are 12,271 annotated facial images in the training set and a total of 3,068 annotated images in the testing set. To establish a baseline for RAF-DB, 2,000-dimensional deep features were extracted using standard CNNs such as VGG [47], AlexNet [12], and DLP-CNN (reviewed in the previous section). The best performing baseline methods achieve an accuracy of 84.13% and an average accuracy of 74.20%.

2.2.2 Wild Facial Expression Recognition methods

Face occlusion is a prevalent issue with wild FER datasets. Facial images might be blocked with hair, glasses, hands, food, or other external objects in some regions and cause occlusion. Empirically, partially occluded faces will downgrade the performance of classification algorithms. Hence, auxiliary strategies need to be designed to circumvent the occlusion issue in FER.

Li *et al.* [48] propose a Patch-based CNN with Attention mechanism (pACNN) to

tackle the occlusion challenge. Intuitively, pACNN focuses only on the informative regions of the input image by assigning large weights to features that are extracted from non-occluded parts of the input, and assigns small weights to features that are extracted from the occluded parts with less information. The weights are automatically learned through an attention mechanism to measure the obstructed-ness of local regions in the input image. pACNN is an end-to-end deep neural network that is composed of two major modules: 1. region decomposition, and 2. occlusion perception.

In the pre-processing step, the input facial image is processed with a face alignment method [49] that is robust to face occlusions, and the corresponding facial landmarks are extracted. 24 individual landmarks that cover informative regions of the face (*e.g.,* two eyes, nose, mouth, and cheek) are chosen to guide the region decomposition module. In forward-propagation step, the input image is fed to a VGG-16 network [47] to yield a convolution feature map of size $512\times28\times28$. The region decomposition module, extracts local feature maps of distinct regions from the network's last convolution feature map. Feature patches of size $512\times6\times6$ are cropped from the final convolution feature map using the position of the 24 significant landmarks as their corresponding patch center. The occlusion perception module then propagates each patch through a Patch-Gated Unit (PG-Unit) to extract local feature vectors for each patch and a corresponding scalar weight to indicate region importance. PG-Unit is composed of feature extraction layers (convolutional and fully-connected layers) and an attention module. The attention module estimates a weight in the range of $[0, 1]$ to be multiplied by the extracted local feature vector. The weighted feature vectors are concatenated to represent the occluded input image. Finally, softmax loss classifies the resulting deep features into six basic facial expression categories and neutral.

A global-local-based CNN with attention mechanism (gACNN) [50] is proposed later to improve on pACNN. gACNN introduces a Global-Local attention method to preserve global information by concatenating weighted global features with weighted local features. Specifically, the final convolution feature map of size $512\times28\times28$ is propagated through a Global-Gated unit (GG-Unit) to extract global features and estimate a global weight. GG-

Unit extracts a global feature vector and estimates a global weight with an attention module similar to pACNN. The final weighted global feature vector is concatenated with individual local feature vectors to represent the occluded input image. Intuitively, the Global-Local representation encodes global and local obstructed-ness for a better classification.

Both pACNN and gACNN are trained on AffectNet and RAF-DB and several lab-controlled datasets. pACNN classifies AffectNet with an accuracy of 55.33%, and classifies RAF-DB with an accuracy of 83.27%. Improved results are achieved with gACNN classifying AffectNet with an accuracy of 58.78%, and classifying RAF-DB with an accuracy of 85.07% across six basic facial expressions and *neutral*.

To create a balance between significant and insignificant facial deep features for FER, Zhao *et al.* [51] introduces a Feature Selection Network (FSN) that automatically preserves significant features and filters out insignificant features. FSN has three modules: 1. Feature extraction module with AlexNet, 2. Feature selection mechanism, and 3. classification. The facial features extracted with AlexNet are passed to the feature selection mechanism that includes two sub-modules. The first sub-module calculates the local influence of extracted features with three stacked convolutional layers and yields a filter mask that filters out irrelevant features for the subsequent layers. The second sub-module, creates a face mask with the same size as the feature map to mask out the features corresponding to the regions that are beyond the face area (*e.g.,* hair, neck, and background). These two generated masks are concurrently multiplied by the feature map to yield the final refined feature map for the classification layers. FSN is trained and evaluated on RAF-DB and FER2013 yielding accuracies of 67.6% and 72.46%, respectively.

Annotating facial expression is a tedious and challenging tasks. On the other hand, deep neural networks require massive annotated datasets to yield satisfactory results. Motivated by this, Florea *et al.* [52] combines semi-supervised learning and inductive transfer learning into an Annealed Label Transfer (ALT) framework to tackle the challenge. ALT, trains a learner network on a labeled FER dataset (RAF-DB and FER+ [53]) and transfers the knowledge to the unlabeled dataset MegaFace [54] to generate pseudo labels. A dual

dataset loss minimizes the error for the labeled dataset (initial knowledge) and increases the confidence of generated pseudo labels for the unlabeled dataset (enhanced knowledge). Since the data generating distribution between the labeled dataset and the unlabeled dataset is assumed to be different, ALT injects randomization by means of an annealing process. The annealing process continuously evaluates the initial knowledge and the enhanced knowledge on the validation based on a temperature (threshold) to randomly preserve the enhanced knowledge. Intuitively, ALT randomly modifies the classification boundaries in the semi-supervised domain to regularize the knowledge transfer. ALT classifies RAF-DB with an accuracy of 84.5% (mean accuracy of 76.5%), and classifies FER2013 and FER+ with accuracies of 69.85% and 85.2%, respectively.

Besides the challenging nature of annotating FER datasets, annotation error and bias is exhibited among popular FER datasets. This can lead to the prediction models that are biased to certain expressions. Additionally, the aggregation of multiple datasets will lose its benefit while the perception of a certain expression such as *fear* is different among different annotators. Zeng *et al.* [55] propose a 3-step framework called Inconsistent Pseudo Annotations to Latent Truth (IPA2LT) to address this issue. In the first step, two prediction models are trained on AffectNet (model A) and RAF-DB (model B). In the second step, the trained models are cross-validated on AffectNet and RAF-DB, and evaluated on an unlabeled dataset (the unlabeled images from AffectNet). The cross dataset predictions are recorded besides the true labels (human annotated) for AffectNet and RAF-DB. The unlabeled dataset is tagged with the predictions from both models. The two predictions for the samples from each dataset are assumed to be inconsistent. In the third step, a Latent Truth Network (LTNet) is trained to extract the latent true label for each sample based on the inconsistencies between the recorded labels. Specifically, a latent label is first estimated by the LTNet and is projected to different models' prediction domain with a transition matrix. The discrepancy between the projected predictions and the recorded predictions are minimized to estimate a more accurate latent truth. IPA2LT is evaluated on RAF-DB, AffectNet, and several other lab-controlled datasets. IPA2LT classifies RAF-DB with an

accuracy of 86.77% and classifies AffectNet with an accuracy of 55.11%.

Lee *et al.* [56] claims that the context around face area can be used in conjunction with the face to boost recognition. Intuitively, the subjects around the person (background, other people, and objects) can contribute to the expressed emotion. The authors propose a Context-Aware Emotion Recognition Network (CAER-Net) to recognize facial expressions in the wild settings. Specifically, the authors have collected a large-scale dataset of video frames from popular TV shows annotated with the facial expression of the person in the frame. CAER-Net consists of two sub-modules: 1. Two-stream encoding networks, and 2. adaptive fusion network. The two stream encoding network encodes the visual features of the extracted face from the video frame and the visual features of the video frame with the face masked out. The latter contributes to the visual cues in the frame context. Additionally, the context features are weighted based on an attention module to filter out the irrelevant features corresponding to the subjects in the scene that do not contribute to the prediction of the facial expression. The extracted features are passed through the adaptive fusion network to be fused for final prediction. The adaptive fusion network automatically weights the two extracted features (facial and context) and encodes them into a single feature vector for classification layer. The authors' developed dataset (CAER) is classified with an accuracy of 77.04%. Additionally, CAER-Net is evaluated on the AFEW [57] dataset with an accuracy of 43.12%.

CHAPTER 3
DISCRIMINANT DISTRIBUTION-AGNOSTIC LOSS FOR FACIAL EXPRESSION RECOGNITION IN THE WILD

3.1 Introduction

While ubiquitous raw large-scale datasets have advanced research in Facial Expression Recognition (FER), two major obstacles hinder the learning performance of deep Convolutional Neural Networks (CNNs) applied in this setting: 1) Large intra-class variation and inter-class similarity, and 2) extreme class imbalance. Due to the in-the-wild attribute, large-scale facial expression datasets acquired in an unconstrained environment inherently populate expression categories with significant variations in pose, gender, age, demography, image quality, and illumination. Additionally, facial expression categorization exhibits an intrinsic imbalance, a prevalent issue in many real-world data [58]. Commonly, categories such as *fear* and *disgust* are minority classes due to lack of representative data. Other expressions such as *neutral*, *happy*, *sad*, *surprise*, and *anger* are majority classes, which are represented with fair amount of data. The data complexity, along with extremely skewed class distribution, can severely degrade the performance of recognition models with deep CNNs.

Extracting discriminative facial features in the embedding space is a critical step towards solving the aforementioned issues. However, the widely used softmax loss is insufficient for delivering discriminant features for classification [59], [60]. Our work is motivated by Wen *et al.* [40], who pioneered center loss as a metric learning approach to yield discriminative deep features by clustering features in the embedding space. Empirically, as illustrated in Figure 3.1 (a), when a CNN model is supervised by center loss in a wild dataset setting, minority classes tend to have overlapping feature clusters. Therefore, recognition performance for minority classes is sub-optimal when deep features learned by center

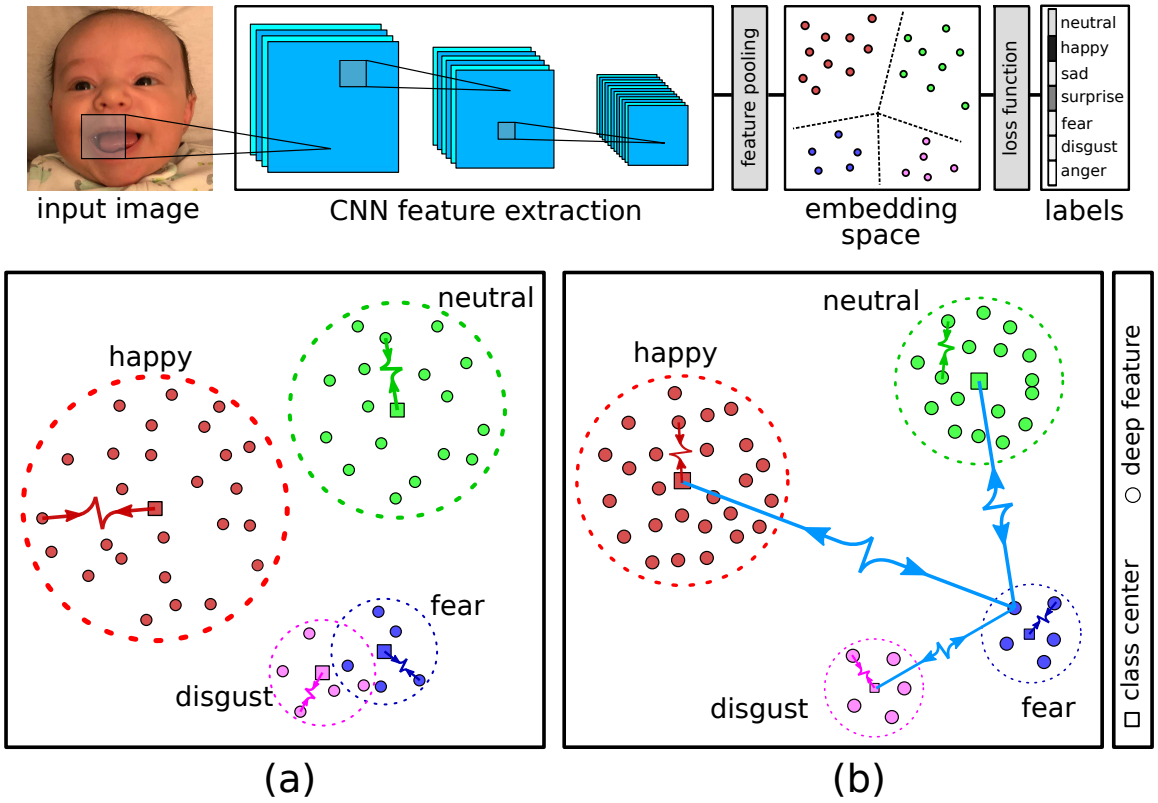


Fig. 3.1: ***Top row:*** Illustration of the general pipeline for FER using a CNN model: CNN features are pooled in the embedding space and a loss function maps the deep features to expression labels. ***Bottom row:*** Example 2-D deep features in the embedding space learned by: (a) Center loss. (b) Discriminant Distribution-Agnostic (DDA) loss. DDA loss pushes the features of a class away from other class centers and pulls them toward their corresponding class centers to create compact and well separated feature clusters for both majority and minority classes.

loss are mapped to expression labels. Due to the inherent complex attributes of a wild FER dataset, optimal recognition of facial expressions requires designing new algorithms to translate raw data into an efficient representation for learning algorithms. To learn discriminative features for FER in the wild, we propose a novel loss function, called Discriminant Distribution-Agnostic loss (DDA loss) to regulate deep features in the embedding space, where extreme class imbalance exits. The CNN models are trained under the joint supervision of softmax loss, center loss, and the proposed DDA loss. As shown in Figure 3.1 (b), DDA loss creates distinctly segregated feature clusters and properly separates both majority and minority classes. Intuitively, DDA loss pushes the features of one class away

from the centers of other classes and pulls them toward their corresponding class center. The discriminant deep features learned using the supervision of the DDA loss are compact and optimally separated in a d-dimensional embedding space. Consequently, the mapping from the embedding space to the label space is more efficient.

Our main contributions are summarized below:

1. We propose a novel loss function called Discriminant Distribution-Agnostic loss (DDA loss) to regulate the distribution of deep features in a d-dimensional embedding space. The proposed DDA loss implicitly maximizes the inter-class separation and minimizes intra-class variations of deep features for both majority and minority classes in extreme class imbalance scenarios. Deep CNNs trained with joint supervision of softmax loss, center loss, and DDA loss yield highly discriminant deep features for wild FER applications.

2. We show that DDA loss can be trained using the standard Stochastic Gradient Descent (SGD) algorithm and can therefore be promptly applied to any state-of-the-art network architectures with minimal intervention.

3. We conduct extensive experiments on a synthesized wild dataset and two popular large-scale wild FER datasets (AffectNet [28] and RAF-DB [27]) to demonstrate the improved recognition results with the proposed method.

3.2 Proposed Method

In this section, we first review necessary preliminaries. We then introduce the proposed Discriminant Distribution-Agnostic loss (DDA loss). Finally, we discuss DDA loss optimization and derive its corresponding gradients in backpropagation for Stochastic Gradient Descent (SGD) optimization.

3.2.1 Preliminaries

Given a training batch of m samples for a K-class image classification problem, let $x_i \in \mathbb{R}^d$ be the output d-dimensional deep feature of the i-th sample belonging to the y_i-th

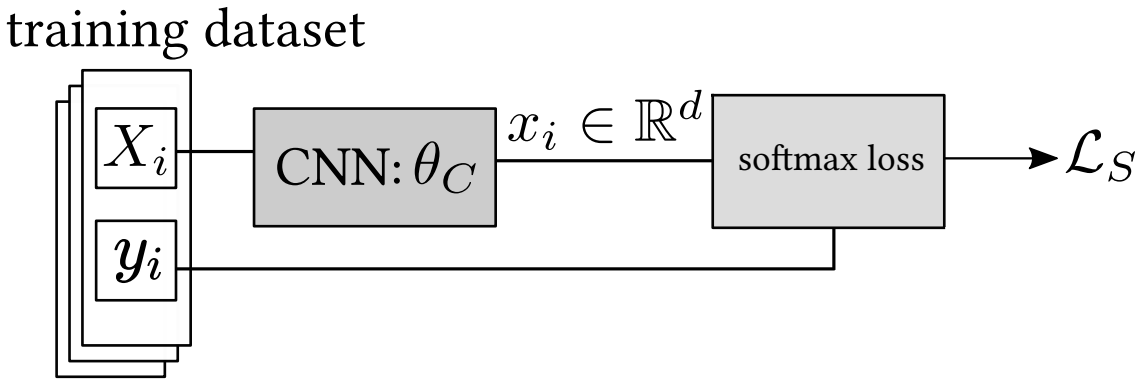


Fig. 3.2: The flow of data in a learning algorithm supervised by softmax loss. CNN extracts the d-dimensional deep features from the input image X_i. The deep feature along with its associated label from the training dataset is fed to the softmax loss function and the scalar value $\mathcal{L}_S$ is calculated.

class, where $y_i \in \{1, ..., K\}$. The conventional softmax loss combines the last fully-connected layer, the softmax function, and the cross-entropy loss to measure the prediction error of the classifier. The last fully connected layer takes x_i and transforms it into a raw score vector (*i.e.,* , logits) $z_i = [z_{i1}, z_{i2}, ..., z_{iK}]^T \in \mathbb{R}^{K\times 1}$ through a linear transformation as follows:

$$z_i = W^T x_i + B \tag{3.1}$$

where $W = [w_1, w_2, ..., w_K] \in \mathbb{R}^{d\times K}$ and $B = [b_1, b_2, ..., b_K] \in \mathbb{R}^{K\times 1}$ are the class weights and bias parameters for the last fully-connected layer, respectively. Each w_j is a d-dimensional vector and each b_j is a scalar where $j \in \{1, ..., K\}$. A probability distribution $p(y = j|x_i) = \frac{e^{z_{ij}}}{\sum_{j=1}^{K} e^{z_{ij}}}$ is then calculated over all classes using the softmax function. Finally, the cross-entropy computes the discrepancy between prediction and ground-truth to formulate the softmax loss function $\mathcal{L}_\mathcal{S}$ as follows:

$$\begin{aligned} \mathcal{L}_S &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{K} y_i \log p(y = j|x_i) \\ &= -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{K} e^{w_j^T x_i + b_j}} \end{aligned} \tag{3.2}$$

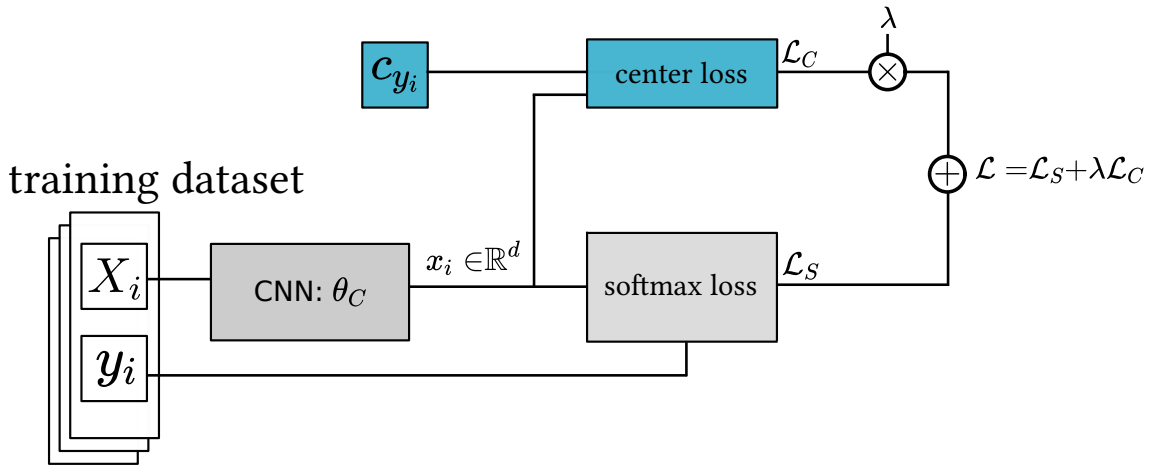


Fig. 3.3: The flow of data in a learning algorithm supervised jointly by softmax loss and center loss. CNN extracts the d-dimensional deep features from the input image X_i. The deep feature along with its associated label from the training dataset is fed to the softmax loss function and the scalar value $\mathcal{L}_S$ is calculated. On the other hand, the deep feature and its corresponding class center is fed to the center loss to calculate the scalar value $\mathcal{L}_C$. Finally, a fraction of center loss (controlled by hyper-parameter λ) is added to the softmax loss.

where m is the total number of samples in a mini-batch. The softmax loss function is minimized by SGD to optimize the network parameters for a better classification. It also makes the learned features separated in an angular fashion in the embedding space since it calculates the vector dot product of $w \cdot x$ to minimize the angle between the deep feature x_i and its corresponding class weight w_{y_i} [59]. The flow of data in a learning algorithm that is supervised by softmax loss is shown in Figure 3.2.

Center loss is jointly optimized with softmax loss to minimize the intra-class variations by minimizing the distance of the deep features to their corresponding class center in a d-dimensional embedding space. The center loss objective function penalizes the Euclidean distance between the deep feature vector of each sample $x_i \in \mathbb{R}^d$ and its corresponding class center $c_{y_i} \in \mathbb{R}^d$ as follows:

$$\mathcal{L}_C = \frac{1}{2m} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \tag{3.3}$$

where y_i is the class that x_i belongs to. Its joint optimization with softmax loss $\mathcal{L}_S$ is given as follows:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C \tag{3.4}$$

where λ controls the contribution of $\mathcal{L}_C$ to the total loss $\mathcal{L}$. Individually, softmax loss $\mathcal{L}_S$ induces inter-class angular separation [61] and center loss $\mathcal{L}_C$ minimizes intra-class Euclidean distances to create compact clusters of features in the embedding space. The softmax loss in Equation 3.2 is a special case of center loss with $\lambda = 0$. The flow of data in a learning algorithm supervised jointly by the softmax loss and center loss is shown in Figure 3.3.

3.2.2 Discriminant Distribution-Agnostic Loss

Training under the joint supervision of softmax loss and center loss creates compact clusters of deep features separated in an angular fashion. The softmax loss formulation incorporates all class weights to emphasize the angular separation of the deep feature x_i and class weights W. However, it has been proven to be unsuitable in a class imbalance setting [62]. On the other hand, center loss only penalizes the distance between a deep feature and its corresponding class center and disregards the contribution of other class centers. In an extreme class imbalance scenario, data points from minority classes and their corresponding class centers are minimally sampled in a training batch. The minimal learning impact from minority classes during mini-batch SGD optimization develops a bias towards majority classes. Thus, the efficiency of a learning algorithm supervised by center loss highly relies on the distribution of data among classes. Notably, in a wild setting, center loss delivers a sub-optimal classification performance for minority classes.

To circumvent this shortcoming, we aim to properly separate clustered deep feature vectors for both minority and majority classes in the embedding space. We argue that the Euclidean distance between the deep feature and all class centers should impact the forward propagation for a single sample to mitigate the bias toward majority classes as evidenced in center loss. To this end, we propose Discriminant Distribution-Agnostic loss (DDA loss)

$\mathcal{L}_{DDA}$ as follows:

$$\begin{aligned}\mathcal{L}_{DDA} &= -\frac{1}{2m}\sum_{i=1}^{m}\sum_{k=1}^{N_k} y_{ik}\log p_C(x_i \in C_k|k) \\ &= -\frac{1}{2m}\sum_{i=1}^{m}\log\frac{e^{-\left\|x_i-c_{y_i}\right\|_2^2}}{\sum_{k=1}^{N_k} e^{-\|x_i-c_k\|_2^2}}\end{aligned} \tag{3.5}$$

where N_k is the number of classes, $y_{ik} = 1$ if x_i belongs to the k-th class and 0 otherwise, and C_k is the cluster for the k-th class in the embedding space. DDA loss estimates the probability of a deep feature x_i belonging to cluster k with its corresponding center c_k using a softmax function. Minimizing $\mathcal{L}_{DDA}$ is equivalent to maximizing the log-likelihood of the estimated probability $p_C(x_i \in C_k|k)$ over a batch of m samples. Compared to softmax loss in Equation 3.2, which emphasizes the angular similarity, DDA loss separates the class features based on the Euclidean distance metric, which correlates with the feature vector's magnitude. Considering the magnitude difference between the learned features of a minority class and a majority class in a class imbalance setting is crucial in achieving intra-class compactness and inter-class separation.

Unlike center loss, DDA loss implicitly pushes the deep feature x_i away from any clusters C_k with $k \neq y_i$ and pulls itself towards its cluster C_k with $k = y_i$ in the embedding space with a single formulation. Intuitively, $\mathcal{L}_{DDA}$ considers the contribution from all majority and minority classes to update network parameters to achieve intra-class compactness and inter-class separability. The proposed DDA loss is distribution-agnostic and mitigates the bias towards majority classes.

DDA loss is jointly optimized with softmax loss and center loss to compose the total loss $\mathcal{L}$ by:

$$\mathcal{L} = \mathcal{L}_S + \lambda\mathcal{L}_C + \gamma\mathcal{L}_{DDA} \tag{3.6}$$

where the hyper-parameter γ controls the contribution of $\mathcal{L}_{DDA}$ to the total loss $\mathcal{L}$ and enables us to conduct quantitative analysis. The center loss defined in Equation 3.4 is considered as a special case of this joint optimization when $\gamma = 0$. The flow of data in a

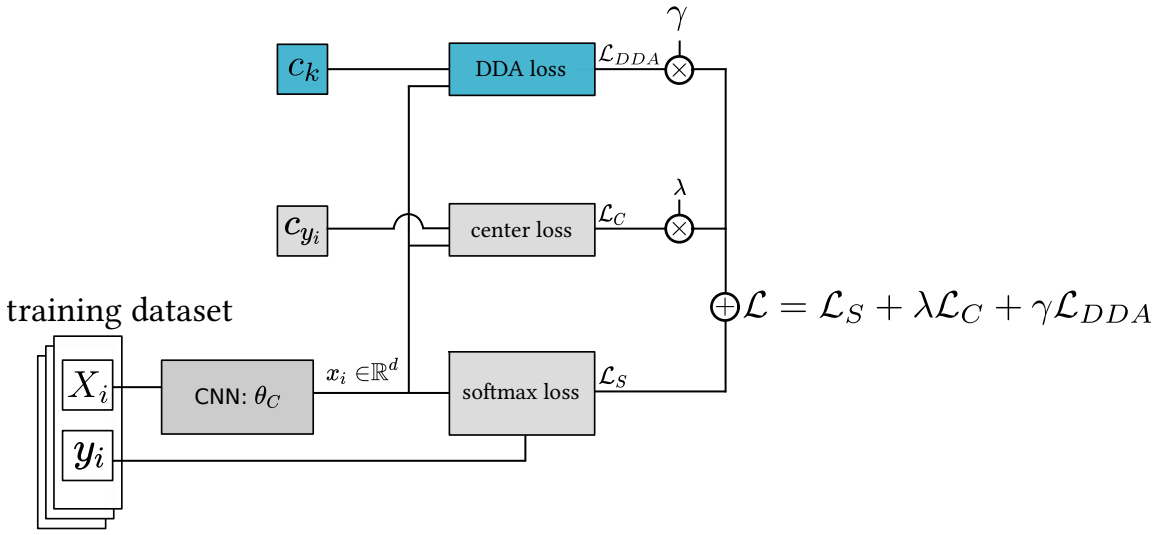


Fig. 3.4: The flow of data in a learning algorithm supervised jointly by softmax loss and center loss. CNN extracts the d-dimensional deep features from the input image X_i. The deep feature along with its associated label from the training dataset is fed to the softmax loss function and the scalar value $\mathcal{L}_S$ is calculated. The deep feature and its corresponding class center is fed to the center loss to calculate the scalar value $\mathcal{L}_C$. Moreover, the deep feature and all the class centers are fed to the DDA loss to calculate the scalar value $\mathcal{L}_{DDA}$. Finally, a fraction of center loss (controlled by hyper-parameter λ) and a fraction of DDA loss (controlled by hyper-parameter γ) are added to the softmax loss.

learning algorithm supervised jointly by the softmax loss, center loss and DDA loss is shown in Figure 3.4.

3.2.3 Optimization

The proposed DDA loss is differentiable and can be optimized with the standard SGD algorithm. We study the SGD back-propagation optimization and the contribution of $\mathcal{L}_{DDA}$ gradients to the total loss $\mathcal{L}$. The joint optimization of $\mathcal{L}_{DDA}$ with softmax loss and center loss contributes to their gradients with respect to the deep feature x_i and centers c_k, respectively.

To simplify the derivative equations, we introduce the following intermediate notation:

$$p_{C_i} = \frac{e^{d_i}}{\sum_{k=1}^{N_k} e^{d_k}} \tag{3.7}$$

where $d_k = -\|x_i - c_k\|_2^2$. The gradient of DDA loss with respect to features x_i are computed according to the chain rule as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{DDA}}{\partial x_i} &= \frac{\partial \mathcal{L}_{DDA}}{\partial d_j} \times \frac{\partial d_j}{\partial x_i} \\ &= -\frac{1}{2m} \times \frac{1}{p_{C_i}} \frac{\partial p_{C_i}}{\partial d_j} \times (-2)(x_i - c_{y_i}) \\ &= \frac{1}{m}(\delta_{ij} - p_{C_j})(x_i - c_{y_i})\end{aligned} \tag{3.8}$$

where the Kronecker delta function is defined as $\delta_{ij} = 1$ for $i = j$ and 0 otherwise.

Class centers are randomly initialized according to the *He* method [18]. We update the centers as follows:

$$c_k = c_k - \alpha \Delta c_k^* \tag{3.9}$$

where Δc_k^* is the combination of an average strategy (Δc_k) proposed in [40] and the gradients of DDA loss with respect to centers c_k as in:

$$\begin{aligned}\Delta c_k^* &= \Delta c_k + \frac{\partial \mathcal{L}_{DDA}}{\partial c_k} \\ &= \frac{\sum_{i=1}^{m} \delta_{y_i k} \cdot (c_k - x_i)}{1 + \sum_{i=1}^{m} \delta_{y_i k}} \\ &\quad + \frac{1}{m} \sum_{i=1}^{m} (\delta_{ij} - p_{C_j})(c_{y_i} - x_i)\end{aligned} \tag{3.10}$$

Algorithm 2 summarizes the major steps for training an end-to-end deep CNN model using DDA loss.

Algorithm 2: The standard mini-batch Stochastic Gradient Descent algorithm for one iteration

Input:

Training dataset $D = \{(X_i, y_i)|i = 1, ..., N\}$;
Mini-batch features $\{x_i|i = 1, 2, ..., m\}$ extracted from a CNN model;
Initialized parameters θ_C for convolutional filters in CNN;
Initialized parameters $W = \{w_j|j = 1, 2, ..., N_k\}$ for the last FC layer;
Initialized centers $C = \{c_k|k = 1, 2, ..., N_k\}$ for center loss and DDA loss;
Hyper-parameters α, γ, λ, and learning rate μ;
The number of iterations $t \leftarrow 0$.

Output: Updated parameters θ_C, W, and C.

while *not converged* **do**

- Compute the total joint loss: $\mathcal{L}^t = \mathcal{L}_S^t + \lambda\mathcal{L}_C^t + \gamma\mathcal{L}_{DDA}^t$.
- Compute the gradients: $\hat{g}_t \leftarrow \frac{1}{m}\sum_{i=1}^{m} \frac{\partial\mathcal{L}_S^t}{\partial x_i^t} + \lambda\frac{\partial\mathcal{L}_C^t}{\partial x_i^t} + \gamma\frac{\partial\mathcal{L}_{DDA}^t}{\partial x_i^t}$.
- Compute Δc_k^* by Equation 3.10.
- $t \leftarrow t + 1$.
- Update w_j for each j: $w_j^{t+1} = w_j^t - \mu\frac{\partial\mathcal{L}_S^t}{\partial w_j^t}$.
- Update c_k for each k: $c_k^{t+1} = c_k^t - \alpha\Delta c_k^{t*}$.
- Update the CNN model parameters θ_C: $\theta_C^{t+1} = \theta_C^t - \mu^t\hat{g}_t$.

3.3 Experiments

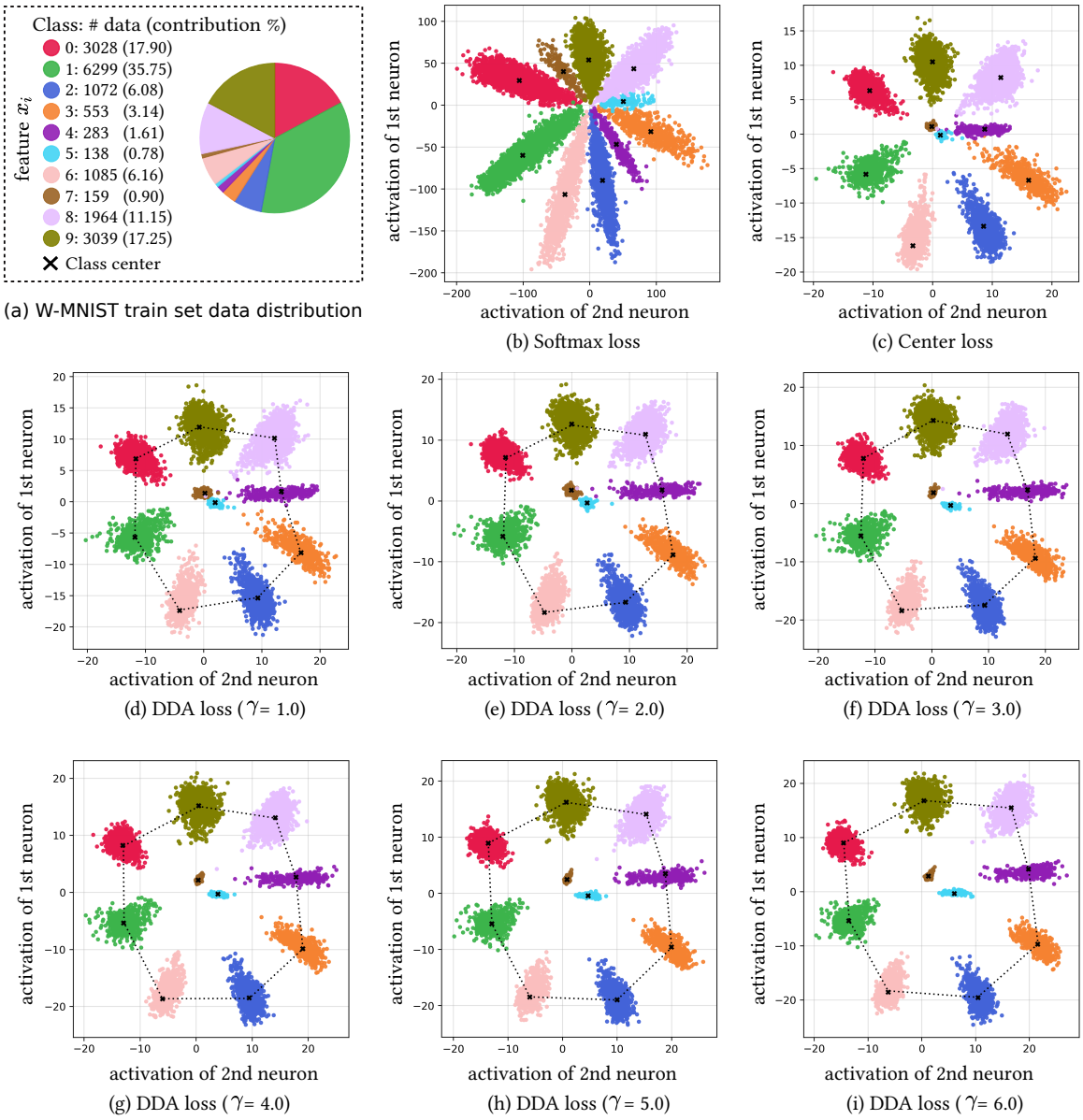
We conduct extensive experiments to evaluate the performance of the proposed loss function and other state-of-the-art methods. We visually and quantitatively validate the superior performance of the proposed Discriminant Distribution-Aware loss (DDA loss) compared to the baseline loss functions, namely, softmax loss and center loss, on a wild toy dataset. We then evaluate the proposed DDA loss on two widely used wild FER datasets against the baseline loss functions and recent state-of-the-art methods that tackle the wild setting.

3.3.1 Wild MNIST Experiments

We present a toy experiment on the Wild MNIST (W-MNIST) dataset with ten classes, a subset of the MNIST dataset [14], to study the proposed method more intuitively. W-MNIST is comprised of randomly sampled image data (single hand-written digits) from the standard MNIST training set. To mimic the characteristics of a wild FER dataset, we drastically decrease the number of training data points in W-MNIST for a few categories by sampling only a few data points from MNIST. The distribution of data in W-MNIST is summarized in Figure 3.5 (a). We illustrate 2D deep features learned by softmax loss and center loss in Figure 3.5 (b) and (c), respectively, and the 2D deep features learned by the proposed DDA loss with different γ values in Figure 3.5 (d)-(i).

To display deep features on a 2D plot, we use the CNN model LeNets++ [40] with six stacked convolutional layers and one fully-connected layer with two neurons. We train LeNets++ on the W-MNIST dataset using the standard SGD with a momentum of 0.9 and a weight decay of 5×10^{-4} for 100 epochs. We use a batch size of 128 and set the initial learning rate as 0.001 with a decay factor of 1.25 every 20 epochs. We do not use any data augmentation on W-MNIST. We empirically set the hyper-parameter λ for center loss as 0.01 and experiment with different γ values for DDA loss.

As illustrated in Figure 3.5, the deep features learned by center loss are more discriminative compared to the deep features learned by softmax loss. However, inter-class distances are optimized with a bias toward the majority classes in the embedding space.



(a) W-MNIST train set data distribution

(b) Softmax loss

(c) Center loss

(d) DDA loss (γ= 1.0)

(e) DDA loss (γ= 2.0)

(f) DDA loss (γ= 3.0)

(g) DDA loss (γ= 4.0)

(h) DDA loss (γ= 5.0)

(i) DDA loss (γ= 6.0)

Fig. 3.5: A wild toy experiment of training LeNets++ on the W-MNIST training set using different loss functions. (a) Distribution of data for the W-MNIST training set. Illustration of the distribution of 2D deep features learned via: (b) Softmax loss, (c) Center loss, (d)-(i) DDA loss with different γ values. It is clear that as the contribution of the DDA loss is increased by increasing the γ value, both majority and minority classes get farther away from each other in the embedding space.

Consequently, minority classes are over-lapped, or their inter-class distances relative to majority classes are not optimized. On the other hand, DDA loss occupies the embedding space with compact and well-separated feature clusters for both majority and minority classes.

Method	λ / γ	Accuracy (%)
softmax loss	-	96.78
center loss	0.01 / -	97.12
DDA loss	0.01 / 1.0	97.17
DDA loss	0.01 / 3.0	97.17
DDA loss	0.01 / 5.0	97.15
DDA loss	0.01 / 7.0	**97.34**

Table 3.1: Classification accuracy on the MNIST's testing set by training the LeNets++ model with different loss functions on the W-MNIST training set.

As we increase the hyper-parameter γ, feature clusters tend to disperse further away from other clusters. Visualization of 2D deep features verifies that the proposed DDA loss yields more discriminative features in a wild dataset setting since it achieves optimal inter-class separation and intra-class compactness for all classes.

To quantitatively evaluate the performance of the proposed DDA loss and the baseline loss functions, we train LeNets++ on the W-MNIST training set and test its recognition performance on the MNIST testing set. Table 3.1 summarizes the classification accuracy of the proposed DDA loss and two baseline loss functions (softmax loss and center loss) on the MNIST testing set. It clearly shows that the proposed DDA loss with $\gamma = 7.0$ outperforms both softmax loss and center loss by achieving an accuracy of 97.34%.

3.3.2 Wild Facial Expression Recognition Experiments

Real-world Affective Face Data-Base (RAF-DB) [27] and AffectNet [28] are the two largest and widely used wild Facial Expression Recognition (FER) datasets. RAF-DB contains 12,271 training images and 3,068 testing images aligned and annotated with six basic expressions (*i.e.,* , *happy*, *sad*, *surprise*, *anger*, *disgust*, and *fear*) and *neutral* expression using crowd-sourcing techniques. The distribution of data in RAF-DB is shown in Figure 3.6. AffectNet contains 280,000 training images and 3,500 testing images manually annotated with six basic expressions and *neutral* expression. The distribution of data in RAF-DB is shown in Figure 3.7. Both datasets comprise of facial images in real world with various gender, age, demography, image quality, and illumination attributes. We first present the

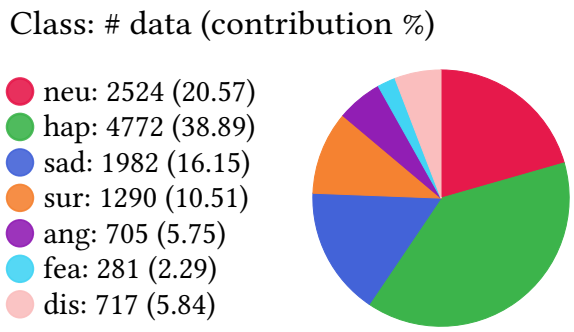


Fig. 3.6: The distribution of data across all classes in RAF-DB. Each color represents a class.

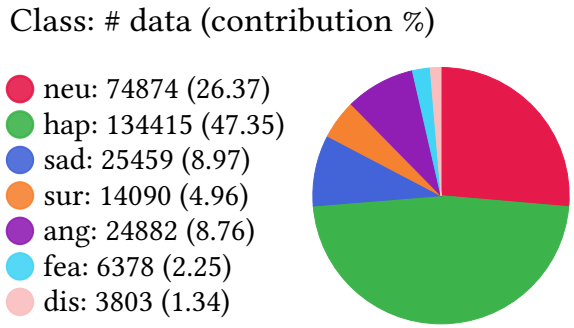


Fig. 3.7: The distribution of data across all classes in AffectNet. Each color represents a class.

details of our implementations in terms of architecture, training, and hyper-parameters. We then analyze the recognition performance on both RAF-DB and AffectNet datasets and study the effect of hyper-parameter γ. Finally, we discuss our results and the limitations of the proposed method.

CNN Architecture: Deep Residual Networks

Deep residual learning [63] aims to solve the *degradation* problem associated with deep models [47, 64]. Specifically, the family of Residual Networks (ResNets), incorporate an identity mapping by integrating residual blocks in a very deep network with many layers. Given an input x to a layer with mapping function $\mathcal{F}(.)$ and parameters W, the output of a standard layer is as follows:

$$y = \mathcal{F}(x, \{W\}) \tag{3.11}$$

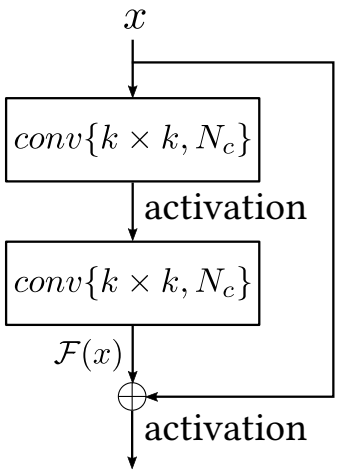


Fig. 3.8: A residual block used in a deep network from the family of deep ResNets. The input x is fed a to a stack of two convolutional layers with filters of size $k \times k$, and N_c channels. The output $\mathcal{F}(x)$ then is added to x through a shortcut path for the block of learning identity mapping.

where y is the output of the layer. Whereas, the output of a residual block is proposed to be:

$$y = \mathcal{F}(x, \{W\}) + x \tag{3.12}$$

Compared to the Equation 3.11, the residual output in Equation 3.12, will learn an identity mapping $y = x$ in case of the vanishing gradient problem that causes the weights W to be zero. The residual operation is implemented by a shortcut connection and element-wise addition, as shown in Figure 3.8. The input x is fed a to a stack of two convolutional layers with filters of size $k \times k$, and N_c channels. The output is then added to x for the block of learning identity mapping.

In practice, each convolutional layer is followed by Batch Normalization (BN) layer [65] and the Rectified Linear Unit (ReLU) activation function [21] as shown in Figure 3.9.

The input in Figure 3.9 is considered to be the same dimension as the output $\mathcal{F}(x)$. If this is not true, a dimension normalization needs to be applied to the input through the shortcut path, as shown in Figure 3.10. Particularly, the input and output dimensions differ when the number of filters N_c changes through the course of stacked layers (by design).

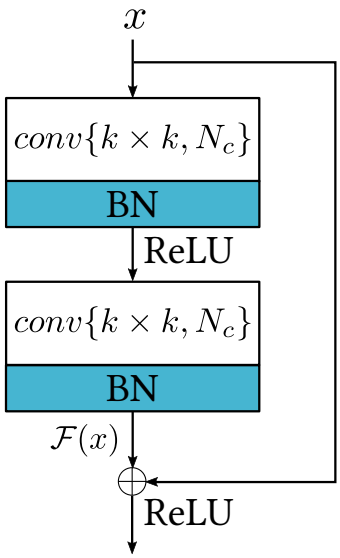


Fig. 3.9: A residual block used in practice. The convolutional layers are followed by the Batch Normalization (BN) layer and the Rectified Linear Unit (ReLU) activation function.

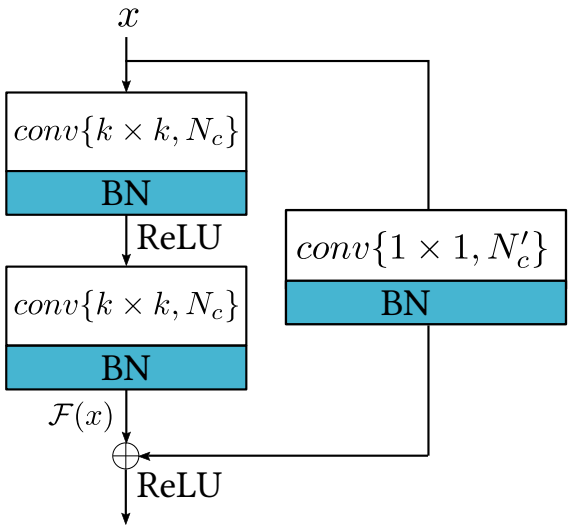


Fig. 3.10: A residual block used in practice when the dimensions of the input x differ from $\mathcal{F}(x)$. Normalization on the input is applied through the shortcut using a convolutional layer with filters of size 1×1 and N'_c channels.

layer name	output size	layer detail
conv1	112×112	$conv\{7 \times 7, 64, 2\}$
maxpool1	56×56	$3 \times 3, stride = 2$
conv2	56×56	$\begin{bmatrix} conv\{3 \times 3, 64\} \\ conv\{3 \times 3, 64\} \end{bmatrix} \times 2$
conv3	28×28	$\begin{bmatrix} conv\{3 \times 3, 128\} \\ conv\{3 \times 3, 128\} \end{bmatrix} \times 2$
conv4	14×14	$\begin{bmatrix} conv\{3 \times 3, 256\} \\ conv\{3 \times 3, 256\} \end{bmatrix} \times 2$
conv5	7×7	$\begin{bmatrix} conv\{3 \times 3, 512\} \\ conv\{3 \times 3, 512\} \end{bmatrix} \times 2$
pooling layer	1×1	average pool, K-neuron fully-connected layer

Table 3.2: The layer details of ResNet-18 for the input of size 224×224. $conv\{k \times k, N_c, s\}$ denotes a convolutional layer with filters of size $k \times k$, N_c channels.

Implementation details

We choose ResNet-18 [63] a standard deep residual network from the family of ResNets due to its close to state-of-the-art performance on canonical visual recognition tasks while offering fewer parameters. Hence, our models are trained faster compared to deeper networks such as VGGs [47] and GoogleNet [66] while maintaining similar recognition performance. ResNet-18 is an 18-layer deep residual CNN as summarized in Table 3.2 for an input of size 224×224 to be classified into K classes. Moreover, the architecture diagram for ResNet-18 is depicted in Figure 3.11.

We fit ResNet-18 to both RAF-DB and AffectNet as the backbone architecture. We train and optimize ResNet-18 using SGD with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay of 5×10^{-4}. Pre-training models with *ImageNet* weights has become a well-established paradigm for many computer vision tasks. However, it has been proven that an *ImageNet* pre-trained model might not always improve the results compared to a model that is trained from scratch [67], especially when the dataset is relatively large. Accordingly, we train ResNet-18 initialized with *ImageNet* weights on RAF-DB, but we train ResNet-18 from scratch on AffectNet since it yields better recognition performance.

Since RAF-DB is significantly smaller than AffectNet, we optimize the learning rate schedule independently for each dataset. We train ResNet-18 on RAF-DB for 60 epochs

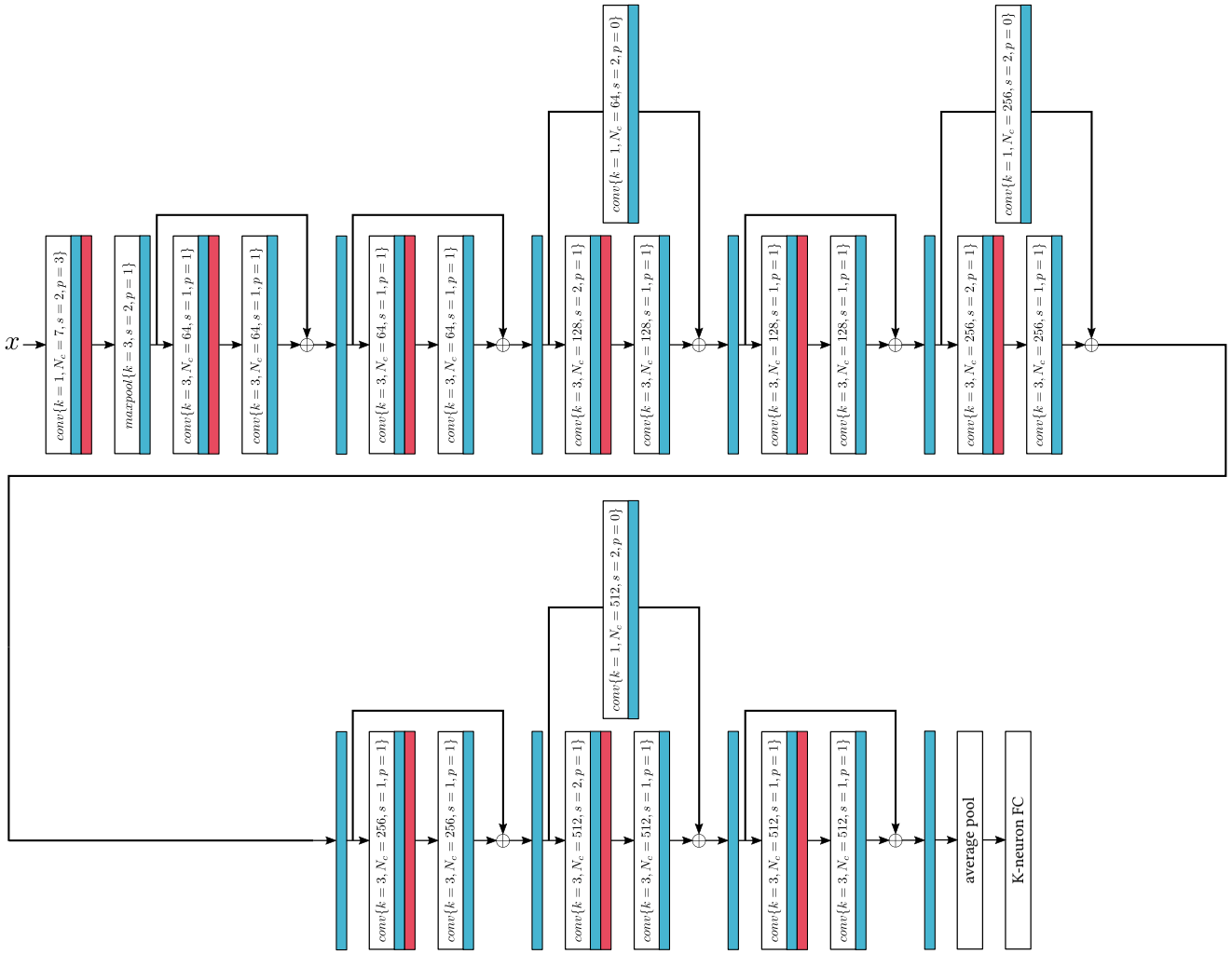


Fig. 3.11: ResNet-18 architecture. k denotes filter size, N_c denotes the number of filters, s denotes the filter stride, and p denotes input padding. Blue layers are Batch Normalization (BN) layers and red layer are Rectified Linear Unit (ReLU) layers.

with a batch size of 64 and decay the learning rate by a factor of 10 every 20 epochs. For AffectNet, we train ResNet-18 for 20 epochs with a batch size of 128 and decay the learning rate by a factor of 10 every five epochs. For both datasets, we augment the input images on-the-fly by extracting random crops (one central, and one for each corner and their horizontal flip). At test time, we use the central crop of the input image. Crops of size 90 (given images of size 100) and 224 (given images of size 256) are extracted from RAF-DB and AffectNet, respectively. Our models are trained using PyTorch deep learning framework [22] on a 2080Ti GPU with 11GBs of V-RAM.

Recognition Performance

Table 3.3 and Table 3.4 compare the expression recognition performance of the proposed DDA loss, the two baseline loss functions, and recent methods on RAF-DB and AffectNet, respectively. Since RAF-DB's testing set is imbalanced, we report both the standard accuracy and the average accuracy, which is the average of the main diagonal values in the confusion matrix. We empirically set the hyper-parameters for center loss as $\lambda = 0.01$ and $\alpha = 0.5$. To ensure a fair comparison, we use these two same hyper-parameters in the proposed DDA loss.

The proposed DDA loss, best optimized with $\gamma = 5.0$ outperforms other methods on RAF-DB by achieving the recognition accuracy of 86.99% and an average recognition accuracy of 79.71%. In terms of the standard accuracy, this is 0.13% improvement on IPA2LT [55] (the best performing state-of-the-art method on RAF-DB), 1.34% improvement on the baseline softmax loss, and 0.65% improvement on the baseline center loss. In terms of the average accuracy, DDA loss achieves an improvement of 2.43% on separate loss [43], 2.43% on the baseline softmax loss, and 1.9% on the baseline center loss.

Similarly, DDA loss, best optimized with $\gamma = 4.0$, outperforms other methods on AffectNet by achieving the recognition accuracy of 62.34%. This is a 3.45% improvement on separate loss [43] (the best performing state-of-the-art method on AffectNet), 0.88% improvement on the baseline softmax loss, and 0.65% improvement on the baseline center loss.

Method	Acc. (%)	Avg. Acc. (%)
FSN [51]	81.10	72.46
pACNN [48]	83.27	-
DLP-CNN [68]	84.13	74.20
MT-ArcVGG [69]	-	76.00
ALT [52]	84.50	76.50
gACNN [50]	85.07	-
separate loss [52]	86.38	77.25
IPA2LT [55]	86.77	-
softmax loss	85.56	77.28
center loss ($\lambda = 0.01$)	86.25	77.81
DDA loss ($\lambda = 0.01$, $\gamma = 5.0$)	**86.90**	**79.71**

Table 3.3: Expression recognition performance of various methods on RAF-DB's testing set in terms of standard accuracy and average accuracy. The top portion of the table lists the results reported in eight state-of-the-art methods while the bottom portion lists our results of the baseline methods (softmax loss and center loss) and the proposed DDA loss, all of which are pre-trained with *ImageNet*.

Method	Accuracy (%)
pACNN [48]	55.33
IPA2LT [55]	57.31
IPFR [70]	57.40
gACNN [50]	58.78
separate loss [43]	58.89
softmax loss	61.46
center loss ($\lambda = 0.01$)	61.69
DDA loss ($\lambda = 0.01$, $\gamma = 4.0$)	**62.34**

Table 3.4: Expression recognition performance of various methods on AffectNet's validation set in terms of accuracy. The top portion of the table lists the results reported in five state-of-the-art methods while the bottom portion lists our results of the baseline methods (softmax loss and center loss) and the proposed DDA loss, all of which are trained from scratch.

It is noteworthy that in our experiments, the margin of improvement in terms of the average recognition accuracy is more significant than the standard accuracy. This is because DDA loss mainly aims to improve the recognition accuracy for the minority classes that achieve poor recognition accuracies in other methods. To further analyze the recognition performance of individual classes, we present the confusion matrices obtained by employing two baseline methods and DDA loss on RAF-DB and AffectNet in Figure 3.12 and Figure 3.13, respectively. It is clear that center loss boosts the recognition rates for most of the majority classes but degrades the recognition rates for minority classes. On the other hand, DDA loss boosts the recognition rates for minority classes and maintains the comparable recognition rates for majority classes. Specifically, we observe that the proposed method either maintains or boosts the recognition rates for majority classes except *neutral* and *surprise* for AffectNet.

In Figure 3.14, we provide sample correctly classified and misclassified images from RAF-DB and AffectNet predicted by our best models trained with DDA loss. Because AffectNet is much larger than RAF-DB, the human annotations are less accurate. This is a prevailing issue for large-scale datasets when resources are low and annotation can be subjective, which leads to more noisy ground-truth labels in AffectNet. Consequently, our models yield correct predictions that might contradict the ground-truth labels.

The Effect of Hyper-parameter γ

Figure 3.15 shows the effect of using different γ values for the proposed DDA loss on the FER performance for wild FER datasets. The contribution of DDA loss to the total loss is controlled by γ. Large γ values make the total loss focus more on DDA loss, and small γ values make the total loss focus more on softmax loss and center loss. Specifically, for large γ values, features either do not separate or do not exhibit compactness in the embedding space. Small γ values cannot separate the feature clusters efficiently to circumvent the issue with the learned features supervised by center loss. Our experiments on two datasets empirically show that softmax loss converges slower and cannot efficiently separate features in angular fashion when increasing the γ value to increase the contribution of DDA loss

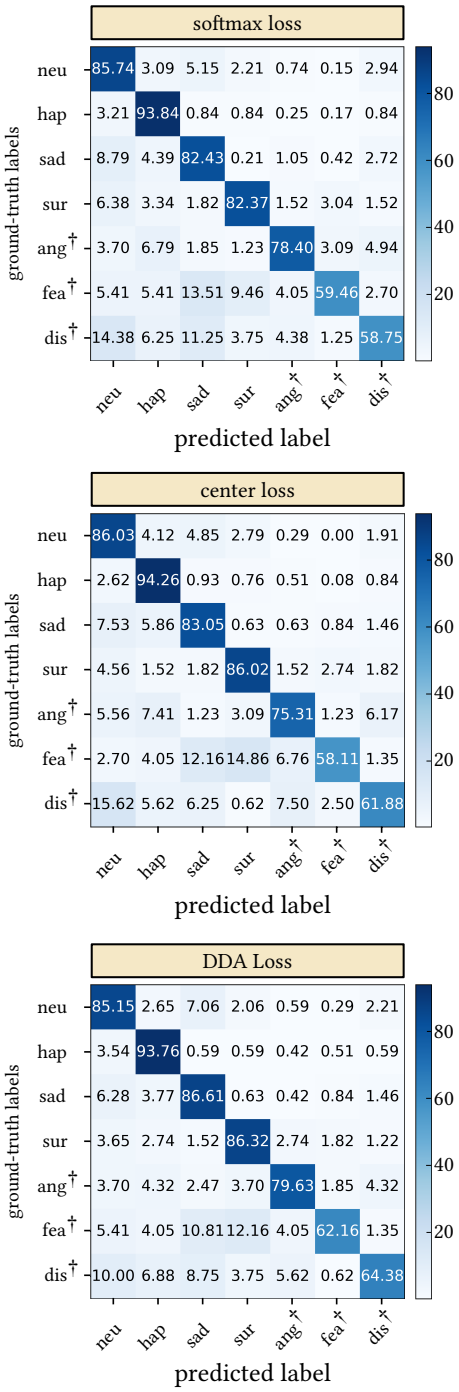


Fig. 3.12: Confusion matrices for the recognition accuracy of RAF-DB using baseline methods and the proposed method. †: Minority classes.

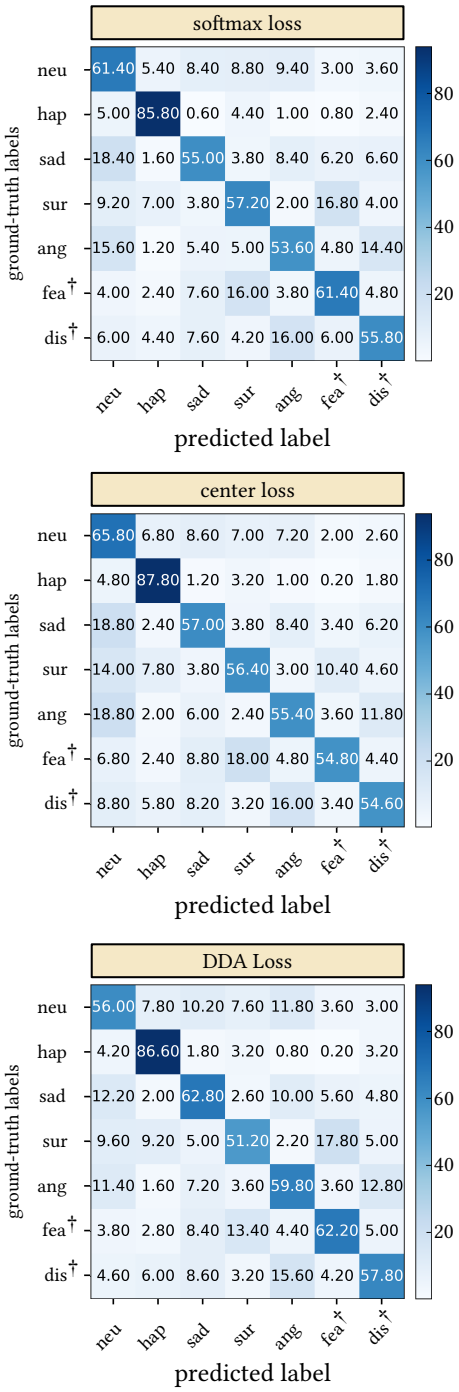


Fig. 3.13: Confusion matrices for the recognition accuracy of AffectNet using baseline methods and the proposed method. †: Minority classes.

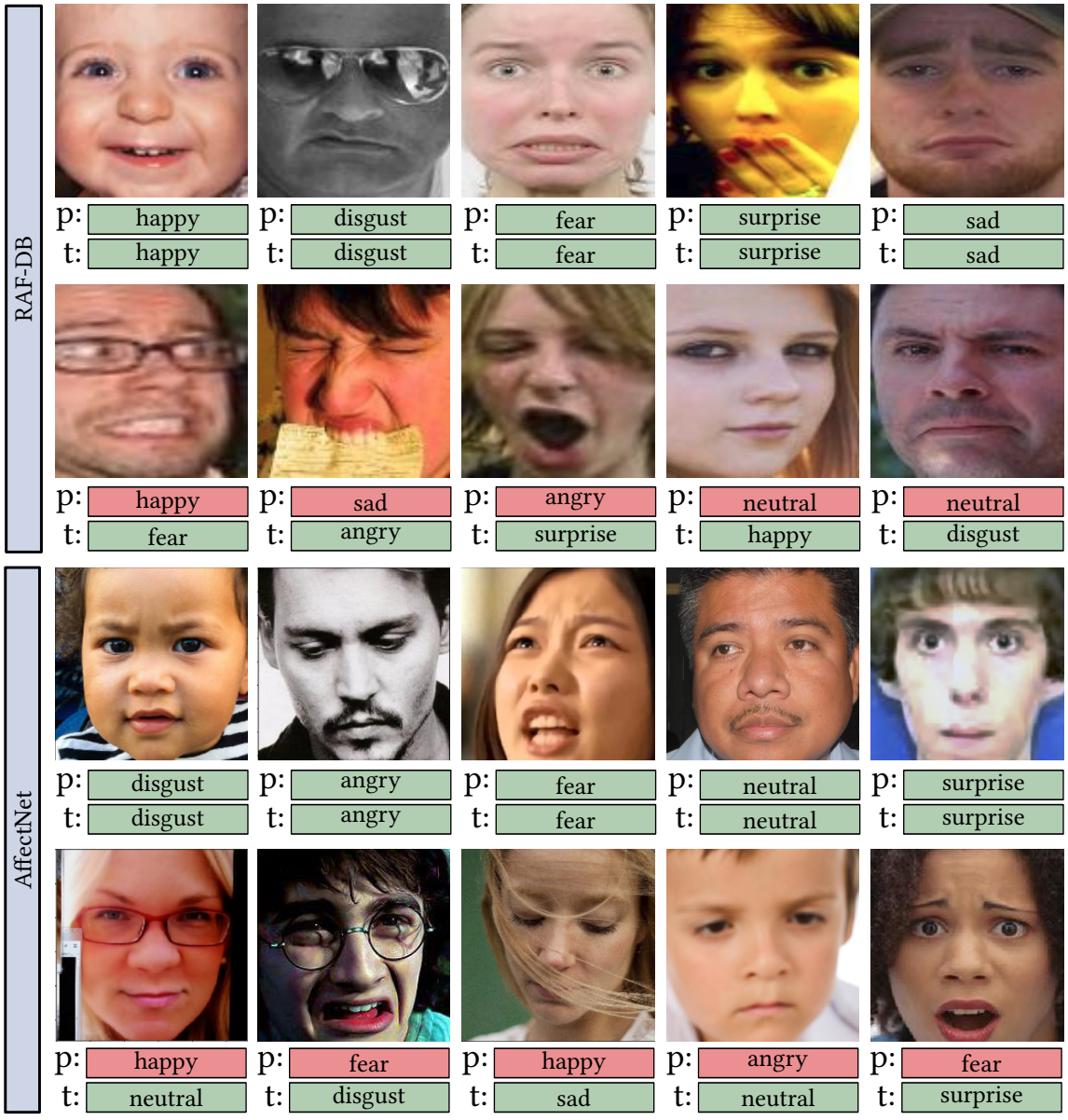


Fig. 3.14: Sample correctly classified and misclassified images in ***top row:*** RAF-DB and ***bottom row:*** AffectNet. ***p*** denotes the predicted label and ***t*** denotes the true label.

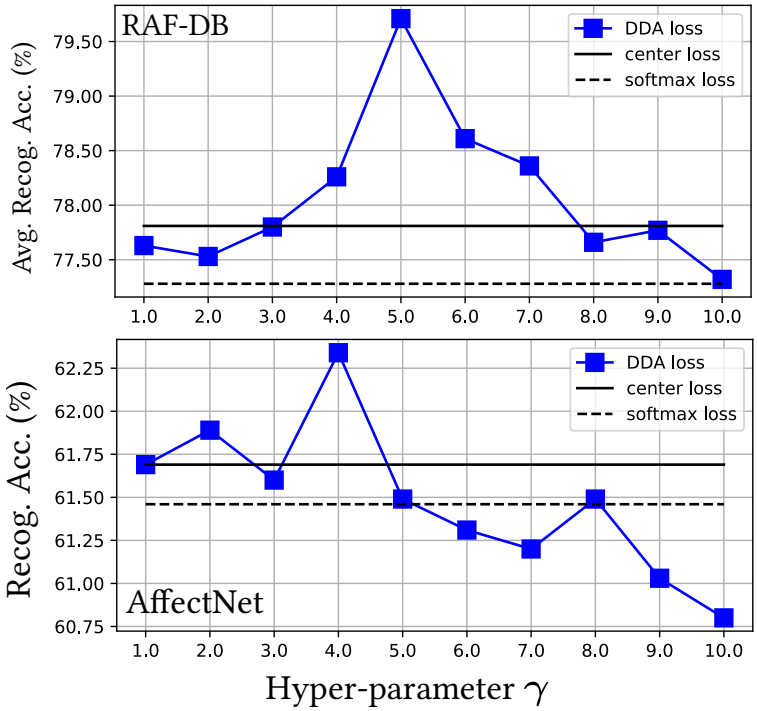


Fig. 3.15: The effect of hyper-parameter γ for DDA loss on (top): The average recognition accuracy of RAF-DB and (bottom): The recognition accuracy of AffectNet.

$\mathcal{L}_{DDA}$ in Equation 3.6. Furthermore, the center loss objective function puts less emphasis on penalizing the distance between features and their corresponding class centers and achieves less intra-class compactness. Hence, the recognition rate starts to degrade after the peak performance with $\gamma = 5.0$ for RAF-DB and $\gamma = 4.0$ for AffectNet. Our experiments show that γ values larger than 10.0 will disrupt the balance between the three terms in the total loss $\mathcal{L}$ in Equation 3.6 and significantly degrade the recognition rates.

Discussion

Although our method boosts the recognition performance when comparing with the two baseline methods, the results are not uniformly positive. For example, the proposed DDA loss tends to outperform center loss for RAF-DB dataset (Figure 3.15 (top)). However, the performance of the proposed method quickly drops below center loss when $\gamma > 4.0$ for AffectNet dataset (Figure 3.15 (bottom)). This behavior is mainly because DDA loss requires to be jointly optimized with center loss to maintain intra-class compactness. Furthermore,

the relative size of majority classes to minority classes in AffectNet is significantly higher than the one in RAF-DB. Consequently, majority classes lose their intra-class structure and the recognition performance drops for all classes when the contribution of DDA loss increases.

3.4 Conclusions

We propose Discriminant Distribution-Agnostic loss (DDA loss) for Facial Expression Recognition (FER) in wild settings. DDA loss implicitly pushes deep features of a class away from other classes and pulls them toward their corresponding class centers in the embedding space. Supervised jointly by softmax loss and center loss, DDA loss efficiently distributes feature clusters of both majority and minority classes in the embedding space where extremely imbalanced distribution of data exists. DDA loss can be optimized with the standard Stochastic Gradient Descent (SGD) algorithm and can be readily employed by any Convolutional Neural Network (CNN) to yield highly discriminative features that are efficient under wild scenarios. Experiments with a synthesized Wild MNIST (W-MNIST) dataset and two widely used wild FER datasets, RAF-DB and AffectNet, demonstrate the superior performance of DDA loss compared to other state-of-the-art methods.

CHAPTER 4
FACIAL EXPRESSION RECOGNITION IN THE WILD VIA DEEP ATTENTIVE CENTER LOSS

4.1 Introduction

In a typical Deep Metric Learning (DML) problem, the deep feature equally contributes to the DML's objective function along all dimensions. Therefore, DML methods are prone to discriminate redundant and noisy information along with important information encoded in the deep feature vector. This leads to over-fitting and hinders the generalization ability of the learning algorithm.

To address the aforementioned, we design a modular attention-based DML approach, called Deep Attentive Center Loss (DACL), to selectively learn to discriminate exclusively the relevant information in the embedding space. Our method is inspired by visual attention described in cognitive neuroscience as the perception of the most relevant subset of sensory data. As shown in Figure 4.1, given the last convolutional spatial feature map as a context, our attention network produces attention weights to guide the DML objective function with the most relevant information. A reformulation of the center loss [40], called sparse center loss, is further proposed as the DML objective function with the advantages of simplicity and straightforward computation. Since our proposed method is designed to be modular, it can be easily developed and integrated with other DML approaches.

The main contributions of our work are summarized as follows:

- We propose a novel attention mechanism that yields attention weights given a context to estimate the weighted contribution of each dimension in the DML's objective function.
- We propose the sparse center loss as the DML's objective function that uses the estimated weights obtained by the attention mechanism to selectively discriminate

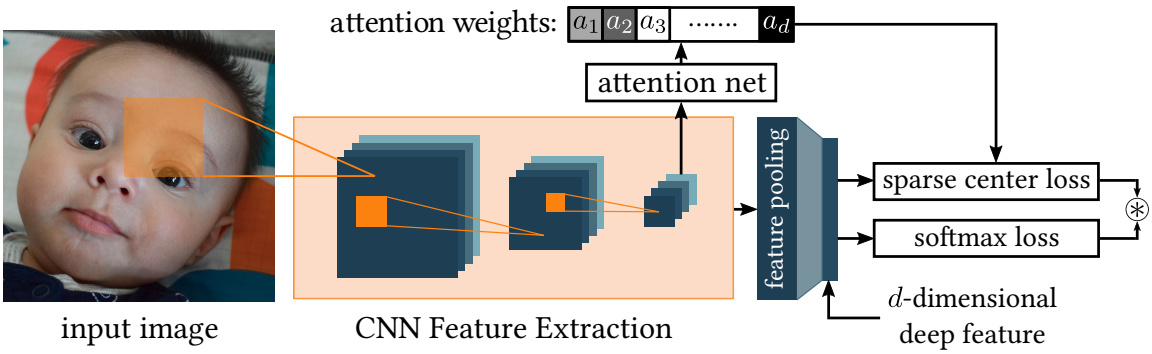


Fig. 4.1: The high-level overview of our proposed Deep Attentive Center Loss (DACL) method: A Convolutional Neural Network (CNN) yields a spatial convolutional features and a feature pooling layer extracts the final d-dimensional deep feature vector for softmax loss and sparse center loss. The last convolutional features are fed to an attention network as context to estimate the attention weights. The estimated weights guide the sparse center loss module to achieve intra-class compactness and inter-class separation for an adaptively selected subset of feature elements. ⊛ indicates a linear combination of softmax loss and sparse center loss.

deep features along its dimensions in the embedding space. Sparse center loss is jointly optimized with softmax loss and can be trained using the standard Stochastic Gradient Descent (SGD).

- We show that the modular DACL method, which consists of the attention network and the sparse center loss, can be trained using the standard Stochastic Gradient Descent (SGD) algorithm and can therefore be promptly applied to any state-of-the-art network architectures and DML methods with minimal intervention.

- We conduct extensive experiments on two popular large-scale wild FER datasets (RAF-DB and AffectNet) to show the improved generalization ability and the superiority of the proposed modular DACL method compared to other state-of-the-arts methods.

4.2 Proposed Method

In this section, we briefly review the necessary preliminaries related to our work. To increase the readability, we rewrite some of the same equations defined in Chapter 3 here.

We then introduce the two building blocks of our proposed Deep Attentive Center Loss (DACL) method, namely, the sparse center loss and the attention network. Additionally, we introduce a gated variant of DACL called g-DACL. Finally, we discuss how DACL is trained and optimized with the standard Stochastic Gradient Descent (SGD).

4.2.1 Preliminaries

Given a training mini-batch of m samples $D_m = \{(X_i, y_i)|i = 1, ..., m\}$, where X_i is the input, and $y_i \in \{1, ..., K\}$ is its corresponding label for a K-class classification problem, let the spatial feature map $x_i^* \in \mathbb{R}^{N_C \times N_H \times N_W}$ be the output of a Convolutional Neural Network (CNN). A pooling layer $\mathcal{P}$ (*e.g.,* fully-connected layers or average pooling layers) takes x_i^* as input and extracts a d-dimensional deep feature $x_i \in \mathbb{R}^d$.

The conventional softmax loss combines a fully-connected layer, softmax function, and the cross-entropy loss to estimate a probability distribution over all classes and measures the prediction error. The deep feature x_i as input to the fully-connected layer is mapped to a raw score vector $z_i = [z_{i1}, ..., z_{iK}]^T \in \mathbb{R}^{K \times 1}$ through a linear transformation as follows:

$$z_i = W^T x_i + B \tag{4.1}$$

where $W = [w_1, ..., w_K] \in \mathbb{R}^{d \times K}$ and $B = [b_1, ..., b_K] \in \mathbb{R}^{K \times 1}$ are the class weights and bias parameters for the fully-connected layer, respectively. A probability distribution $p(y = j|x_i)$ is then calculated over all classes using the softmax function. Finally, the cross-entropy loss function computes the discrepancy between prediction and the true label y_i to formulate the softmax loss function $\mathcal{L}_\mathcal{S}$ as follows:

$$\begin{aligned} \mathcal{L}_S &= -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{K} y_i \log p(y = j|x_i) \\ &= -\frac{1}{m} \sum_{i=1}^{m} \log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{K} e^{w_j^T x_i + b_j}} \end{aligned} \tag{4.2}$$

4.2.2 Sparse Center Loss

Center loss is a widely adopted DML method where the similarity is measured between the deep features and their corresponding class centers (class prototypes). The objective function in center loss minimizes the Within Cluster Sum of Squares (WCSS) between the deep features and their corresponding class centers. That is, it aims to partition the embedding space into K clusters for a K-class classification problem. Given a training mini-batch of m samples, let $\mathbf{x_i} = [x_{i1}, x_{i2}, ..., x_{id}]^T \in \mathbb{R}^d$ be the i-th sample deep feature vector belonging to the y_i-th class, where $y_i \in \{1, ..., K\}$ and $\mathbf{c_{y_i}} = [c_{y_i1}, ..., c_{y_id}]^T \in \mathbb{R}^d$ be its corresponding class center. Center loss minimizes the following criterion defined as:

$$\mathcal{L}_C = \frac{1}{2m} \sum_{i=1}^{m} \sum_{j=1}^{d} \|x_{ij} - c_{y_ij}\|_2^2 \tag{4.3}$$

where WCSS is minimized by equally penalizing the Euclidean distance between the deep features and their corresponding class centers in the embedding space.

We argue that not all the elements in a feature vector are relevant to discrimination. Our goal is to select only a subset of elements in a deep feature vector to contribute in the discrimination. Accordingly, to filter out irrelevant features in the discrimination process, we weight the calculated Euclidean distance at each dimension in Equation 4.3 and develop a a sparse center loss method as follows:

$$\begin{gathered} \mathcal{L}_{SC} = \frac{1}{2m} \sum_{i=1}^{m} \sum_{j=1}^{d} a_{ij} \odot \|x_{ij} - c_{y_ij}\|_2^2 \\ \text{subject to} \quad 0 < a_{ij} \leq 1 \quad \forall j, \quad (j = 1, ..., d). \end{gathered} \tag{4.4}$$

where $\odot$ indicates element-wise multiplication and a_{ij} denotes the weight of the i-th deep feature along the dimension $j \in \{1, ..., d\}$ in the embedding space. Intuitively, the sparse center loss calculates a weighted WCSS. It should be noted that Equation 4.4 reduces to the standard center loss in Equation 4.3 if $a_{i1} = ... = a_{id}$.

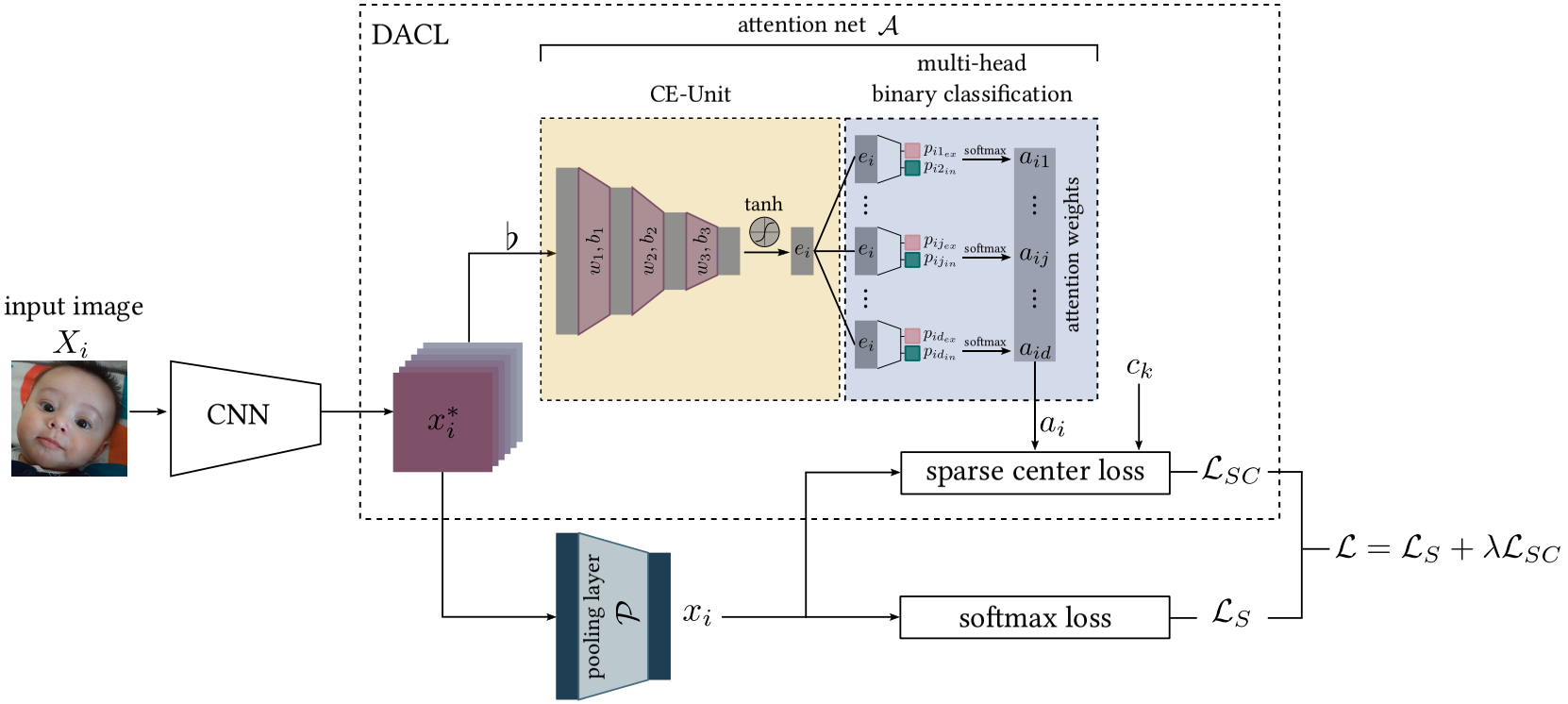


Fig. 4.2: The illustration of the proposed DACL method. An input image X_i is fed to the CNN to yield the convolutional spatial feature map x_i^*. DACL is a hybrid combination of an attention network $\mathcal{A}$ and a sparse center loss. The CE-Unit in DACL's attention mechanism takes the spatial feature map as a context and yields an encoded latent feature vector e_i to eliminate noise and irrelevant information. A multi-head binary classification module then calculates the attention weight a_{ij} corresponding to the j-th dimension in the deep feature x_i at dimension j. Finally, the sparse center loss $\mathcal{L}_{SC}$ calculates a weighted WCSS and is fractionally accumulated with the softmax loss $\mathcal{L}_S$ to compose the final loss $\mathcal{L}$.

4.2.3 Attention Network

We design an auxiliary attention network attached to the CNN to dynamically estimate the weights $a_i \in \mathbb{R}^d$ for the sparse center loss based on the input. Specifically, we seek an adaptive and flexible approach to estimate the weights for the sparse center loss that adjusts to the task and the input data. Ideally, we require the weights to be determined by a neural network. For this purpose, we propose an attention network $\mathcal{A}$ that adaptively computes an attention weight vector to govern the contribution of deep feature x_i along the j-th dimension in Equation 4.4. This attention network together with the sparse center loss comprises the two building blocks of the proposed DACL method. Figure 4.2 presents the proposed attention network in DACL. It has two major components: 1. The Context Encoder Unit (CE-Unit), which takes the spatial feature map from the CNN as input (context) and generates a latent representation and 2. The multi-head binary classification module that takes the latent representation and estimates the attention weights. It should be emphasized that the context for the attention network is at the convolutional feature-level to preserve the spatial information.

We build a dense CE-Unit by stacking three trainable fully-connected linear layers to extract exclusively relevant information from the context as follows:

$$e_i = \tanh(\text{BN}(W_3^T \text{relu}(\text{BN}(W_2^T \text{relu}(\text{BN}(W_1^T \flat(x_i^*) + b_1)) + b_2)) + b_3)) \tag{4.5}$$

where x_i^* is the last convolutional feature map in the CNN *i.e.,* the context feature for the i-th sample, the operator $\flat : \mathbb{R}^{1\times N_C \times N_H \times N_W} \rightarrow \mathbb{R}^{1\times \mathcal{N}_\mathcal{C}\mathcal{N}_\mathcal{H}\mathcal{N}_\mathcal{W}}$ flattens the convolutional feature map, W_l and b_l are respectively the weights and biases for l-th linear layer in the attention network where $l = 1, 2, 3$. Layers are interjected with batch normalization BN(.) [65] and rectified linear units relu(.) to capture non-linear relationships between layers. The final hyperbolic tangent function tanh(.) as element-wise non-linearity preserves both positive and negative activation values for a smoother gradient flow in the network. We initialize the linear layer weights using the *He* initialization method [18], and the biases are initialized to 0. The CE-Unit defined in Equation 4.5 extracts an encoded latent feature

vector $e_i \in \mathbb{R}^{d' \ll d}$ for the i-th sample in a lower dimension to eliminate irrelevant information while keeping the important information. The CE-Unit is adjustable in terms of layer parameters to match a specific task.

To estimate the attention weight of the j-th dimension correlating to the d-dimensional deep feature x_i at dimension j, we attach a multi-head binary classification (inclusion / exclusion) module to the CE-Unit. The latent d'-dimensional feature vector e_i is shared among d linear units *i.e.,* heads with two outputs each, to calculate two raw scores for the deep feature x_i along dimension j as follows:

$$\begin{aligned} p_{ij_{in}} &= A_{j_{in}}^T e_i + b_{j_{in}} \\ p_{ij_{ex}} &= A_{j_{ex}}^T e_i + b_{j_{ex}} \end{aligned} \tag{4.6}$$

where $A_j \in \mathbb{R}^{d' \times 2}$ and $b_j \in \mathbb{R}^2$ are the learnable weights and biases for each classification head with subscript in representing inclusion and subscript ex representing exclusion, and $p_{ij_{in}}$ and $p_{ij_{ex}}$ denote the inclusion and exclusion scores for the j-th dimension in x_i, respectively. A softmax function is applied on each head's output to normalize the scores subject to the constraint in Equation 4.4. Finally, the corresponding attention weight a_{ij} is calculated as follows:

$$a_{ij} = \frac{\exp(p_{ij_{in}})}{\exp(p_{ij_{in}}) + \exp(p_{ij_{ex}})} \tag{4.7}$$

The differentiable softmax function employed on the raw scores will limit the value of the estimated attention weights in the range $(0, 1]$.

4.2.4 Gated CE-Unit

We propose to integrate the gating mechanism [71] in the CE-Unit to learn complex relationships while encoding the context feature. Specifically, we aim to concurrently learn a sigmoidal relationship between the context feature and the final encoded latent feature vector to compensate the approximately linear region in $[-1, 1]$ for hyperbolic tangent function. We name this variant of the proposed method as the gated-DACL (g-DACL). The gated encoding is calculated by forward-propagating the context feature through two stacked

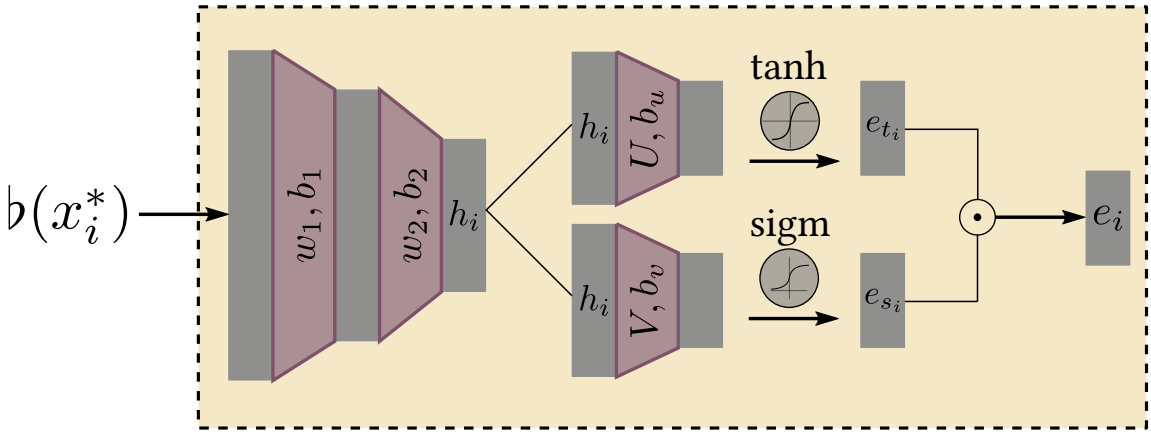


Fig. 4.3: The illustration of gated CE-Unit in g-DACL. The intermediate encoding h_i is calculated with two stacked fully-connected layers. Then, h_i is shared between two separate fully-connected layers to yield two activated encodings e_{t_i} and e_{s_i}. Finally, the encoded latent feature vector is calculated by $e_{t_i} \odot e_{s_i}$.

trainable fully-connected linear layers to yield the intermediate encoding as follows:

$$h_i = \text{relu}(\text{BN}(W_2^T \text{relu}(\text{BN}(W_1^T \flat(x_i^*) + b_1)) + b_2)) \tag{4.8}$$

Then the intermediate encoding h_i is shared between two separate fully-connected layers to yield two activated encodings as:

$$\begin{aligned} e_{t_i} &= \tanh(\text{BN}(U^T e_i + b_u)) \\ e_{s_i} &= \text{sigm}(\text{BN}(V^T e_i + b_v)) \end{aligned} \tag{4.9}$$

where $\{U, V, b_u, b_v\}$ is the set of parameters for the two separate fully-connected layers, and sigm(.) is the sigmoid non-linear function. Finally, the encoded latent feature vector is computed as follows:

$$e_i = e_{t_i} \odot e_{s_i} \tag{4.10}$$

We illustrate the modified gated CE-Unit mechanism in Figure 4.3.

4.2.5 Training and Optimization

Our proposed DACL method as illustrated in Figure 4.2 is trained in an end-to-end

manner where the sparse center loss is jointly supervised with softmax loss to compose the final loss as follows:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_{SC} \tag{4.11}$$

where λ controls the contribution of the sparse center loss $\mathcal{L}_{SC}$ to the total loss $\mathcal{L}$. The parameters associated with DACL can be optimized using the standard SGD algorithm. The gradient of the sparse center loss with respect to the deep features are obtained as follows:

$$\frac{\partial \mathcal{L}_{SC}}{\partial x_i} = \frac{1}{m} a_i \odot (x_i - c_{y_i}) \tag{4.12}$$

and the gradient of the sparse center loss with respect to the attention weights are obtained as follows:

$$\frac{\partial \mathcal{L}_{SC}}{\partial a_i} = \frac{1}{2m} a_i \odot \|x_i - c_{y_i}\|_2^2 \tag{4.13}$$

The centers c_k are initialized using the *He* initialization method and are updated according to a moving average strategy as follows:

$$\Delta_{c_k} = \frac{\sum_{i=1}^{m} \delta_{y_i j} a_i \odot (c_j - x_i)}{\epsilon + \sum_{i=1}^{m} \delta_{y_i j}} \tag{4.14}$$

where the Kronecker delta function is defined as $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. The gradients with respect to the context feature x_i^* is trivially calculated according to the chain rule. We summarize training a supervised learning algorithm (*e.g.,* prediction model) with DACL in Algorithm 3.

4.3 Experiments

In this section, we conduct extensive experiments on two widely used wild Facial Expression Recognition (FER) datasets (*e.g.,* RAF-DB [27] and AffectNet [28]) to demonstrate the superior performance of our proposed Deep Attentive Center Loss (DACL). We evaluate our method on the wild FER datasets compared with two baselines (softmax loss and center loss) and various other state-of-the-art methods. Finally, we visualize the learned attention

Algorithm 3: Training a supervised learning algorithm (*e.g.,* prediction model) with DACL.

Input:

Training dataset $D = \{(X_i, y_i)|i = 1, ..., N\}$;
Initialized CNN parameters θ_C;
Initialized pooling layer parameters θ_P;
Initialized attention network parameters θ_A;
Initialized softmax loss FC layer θ_S;
Initialized centers $C = \{c_k|k = 1, ..., K\}$;
Hyper-parameters α, λ, and learning rate μ;
The number of iterations $t \leftarrow 0$.

Output: Updated parameters θ_C, θ_P, θ_A, θ_S, and C.

while *not converged* **do**

Sample a mini-batch of size m from the training dataset: $D_m = \{(X_i, y_i)|i = 1, ..., m\}$.

Compute the context features using the CNN: $\{x_i^*|i = 1, ..., m\}$.

Compute the deep features using the pooling layer: $\{x_i|i = 1, ..., m\}$.

Compute the attention weights $\{a_i|i = 1, ..., m\}$ by Equations 4.5 - 4.7.

`// Note: Use Equations 4.8 - 4.10 instead of Equation 4.5 for g-DACL.`

Compute the softmax loss $\mathcal{L}_S^t$ by Equation 4.2.

Compute the sparse center loss $\mathcal{L}_{SC}^t$ by Equation 4.4.

Compute the total loss by Equation 4.11: $\mathcal{L}^t = \mathcal{L}_S^t + \lambda \mathcal{L}_{SC}^t$.

Compute the softmax loss gradients: $\hat{g}_S^t \leftarrow \frac{\partial \mathcal{L}_S^t}{\partial \theta_S}$.

Compute the pooling layer gradients: $\hat{g}_P^t \leftarrow \frac{1}{m} \sum_{i=1}^{m} \frac{\partial x_i^t}{\partial \theta_P} \frac{\partial \mathcal{L}_S^t}{\partial x_i^t} + \lambda \frac{\partial \mathcal{L}_{SC}^t}{\partial x_i^t}$.

Compute the attention network gradients: $\hat{g}_A^t \leftarrow \frac{1}{m} \sum_{i=1}^{m} \frac{\partial a_i^t}{\partial \theta_A} (\frac{\partial \mathcal{L}_{SC}^t}{\partial a_i^t})$.

Compute the CNN gradients: $\hat{g}_C^t \leftarrow \frac{1}{m} \sum_{i=1}^{m} \frac{\partial x_i^{*t}}{\partial \theta_C} (\frac{\partial \mathcal{A}^t}{\partial x_i^{*t}} + \frac{\partial \mathcal{P}^t}{\partial x_i^{*t}})$.

Compute Δc_k by Equation 4.14.

$t \leftarrow t + 1$.

Update c_k for each k: $c_k^{t+1} = c_k^t - \alpha \Delta c_k$.

Update the model parameters: $\theta_{\{C,P,A,S\}}^{t+1} = \theta_{\{C,P,A,S\}}^t - \mu^t \hat{g}_{\{C,P,A,S\}}^t$.

weights to interpret our model intuitively.

4.3.1 Implementation Details

We use ResNet-18 [63] a standard Convolutional Neural Network (CNN) as our backbone architecture in our experiments. At the time of writing this chapter, we decided to explore pre-training our models with *MS-CELEB-1M* [72], a face dataset with 10 million images of nearly 100,000 subjects. Since FER's domain is facial images, we argue that FER models pre-trained on a Face Recognition dataset compared to *ImageNet* improves the results. Additionally, *MS-CELEB-1M* is an equivalent of *ImageNet* in the facial image domain in terms of scale and diversity.

We use the standard Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. We augment the input images on-the-fly by extracting random crops (one central, and one for each corner and their horizontal flips). At test time, we use the central crop of the input image. Crops of size 224×224 are extracted from the input images with size 256×256. We train ResNet-18 on RAF-DB for 60 epochs with an initial learning rate of 0.01 decayed by a factor of 10 every 20 epochs. Alternatively, we train ResNet-18 on AffectNet for 20 epochs with an initial learning rate of 0.01 decayed by a factor of 5 every five epochs. We use a batch size of 128 for both datasets. The hyper-parameters α and λ are empirically set as 0.5 and 0.01.

With our specific backbone architecture setup, the deep feature x_i is 512-dimensional, the last convolutional feature map x_i^* is of size $512 \times 7 \times 7$ and the pooling layer is the standard 2D average pooling layer in ResNet-18. The CE-Unit in DACL is designed by stacking three fully-connected layers with 3,584, 512, and 64 channels, respectively. Hence, the latent feature vector e_i is 64-dimensional. Accordingly, we have 512 heads in our multi-head binary classification module that yields a 512-dimensional attention weight vector. We train our models using the PyTorch deep learning framework [22] on an NVIDIA 2080Ti GPU with 11GB of V-RAM.

4.3.2 Recognition Results

We present wild FER results in Table 4.1 and Table 4.2 for RAF-DB and AffectNet, respectively. Since the testing set for RAF-DB is imbalanced, we report the average accuracy, which is the mean of diagonal values in the confusion matrix alongside the standard accuracy across all classes. We only report the standard accuracy on AffectNet, since the validation set is balanced. Because we pre-train our proposed model with a new dataset (*MS-CELEB-1M*) in this chapter, we do the same with our baseline methods and the Discriminant Distribution-Agnostic loss (DDA loss) proposed in Chapter 3 (same hyper-parameters). Hence, two sets of results for softmax loss, center loss, and DDA loss are reported: one with the pre-training paradigm used in Chapter 3 and one with the new pre-training paradigm used in this chapter.

Our DACL method achieves a recognition accuracy of 87.78% and an average recognition accuracy of 80.44% on RAF-DB that outperforms our baseline methods and other state-of-the-art methods. In terms of the standard accuracy, DACL achieves a 0.88% improvement on our best DDA loss method, a 1.24% improvement on the best softmax loss baseline method, and a 0.72% improvement on the best center loss baseline. In terms of the average accuracy, DACL achieves a 0.44% improvement on our best DDA loss method, a 1.01% improvement on the best softmax loss baseline, and a 0.73% improvement on the best center loss baseline. g-DACL with the standard accuracy of 87.19% and the average accuracy of 79.56% delivers comparable results compared to the best center loss method.

Similarly, DACL outperforms the baseline methods and other-state-of-the-art methods on AffectNet with an accuracy of 65.20%. This is a 0.94% improvement on our best DDA loss method, a 1.34% improvement on the best softmax loss baseline, and a 1.11% improvement on the best center loss baseline. We also notice that DACL improves both baseline methods by a larger margin compared to the margin of improvement by center loss over softmax loss. In other words, center loss improves on softmax loss, but the generalization ability is sub-optimal. However, our proposed DACL significantly improves the generalization ability of the center loss. g-DACL delivers an accuracy of 65.17%, comparable to the naive DACL.

Method	Acc. (%)	Avg. Acc. (%)
FSN [51]	81.10	72.46
pACNN [48]	83.27	-
DLP-CNN [27]	84.13	74.20
MT-ArcVGG [69]	-	76.00
ALT [52]	84.50	76.50
gACNN [50]	85.07	-
separate loss [43]	86.38	77.25
IPA2LT [55]	86.77	-
softmax loss [*ImageNet*]	85.56	77.28
center loss [*ImageNet*]	86.25	77.81
DDA Loss [*ImageNet*]	86.90	79.71
softmax loss [*MS-CELEB-1M*]	86.54	79.43
center loss [*MS-CELEB-1M*]	87.06	79.71
g-DACL [*MS-CELEB-1M*]	87.19	79.56
DDA Loss [*MS-CELEB-1M*]	86.83	80.00
DACL [*MS-CELEB-1M*]	**87.78**	**80.44**

Table 4.1: Expression recognition performance of various methods on RAF-DB's testing set in terms of standard accuracy and average accuracy. The top portion of the table lists the results reported in eight state-of-the-art methods while the bottom portion lists our results of the baseline methods (softmax loss and center loss), DDA loss, DACL, and g-DACL. The name within each bracket indicates the dataset that the model is pre-trained with.

Method	Accuracy (%)
pACNN [48]	55.33
IPA2LT [55]	57.31
IPFR [70]	57.40
gACNN [50]	58.78
separate loss [43]	58.89
softmax loss	61.46
center loss	61.69
DDA Loss	62.34
softmax loss [*MS-CELEB-1M*]	63.86
center loss [*MS-CELEB-1M*]	64.09
DDA Loss [*MS-CELEB-1M*]	64.26
g-DACL [*MS-CELEB-1M*]	65.17
DACL [*MS-CELEB-1M*]	**65.20**

Table 4.2: Expression recognition performance of various methods on AffectNet's validation set in terms of accuracy. The top portion of the table lists the results reported in five state-of-the-art methods while the bottom portion lists our results of the baseline methods (softmax loss and center loss), DDA loss, DACL, and g-DACL. The name within each bracket indicates the dataset that the model is pre-trained with. No bracket indicates that the model is trained from scratch.

We depict some correctly classified and misclassified sample images from both wild FER datasets by the DACL method in Figure 4.4.

4.3.3 Discussion

It is clear that pre-training models with the *MS-CELEB-1M* dataset boosts the performance across all methods except for DDA loss on RAF-DB. We observe that DDA loss tends to perform better than baseline methods (softmax loss and center loss) on both datasets regardless of the pre-training paradigm. However, DDA loss pre-trained with *MS-CELEB-1M* achieves worse performance than center loss pre-trained with *MS-CELEB-1M* on RAF-DB. We believe DDA loss will achieve better results than center loss with a carefully chosen γ value. It is noteworthy that DDA loss pre-trained with *MS-CELEB-1M* exclusively improves the average accuracy on RAF-DB compared to the baseline methods pre-trained with *MS-CELEB-1M*. This is because DDA loss focuses on improving the recognition accuracy of minority classes. On the other hand, DACL improves both the standard accuracy and

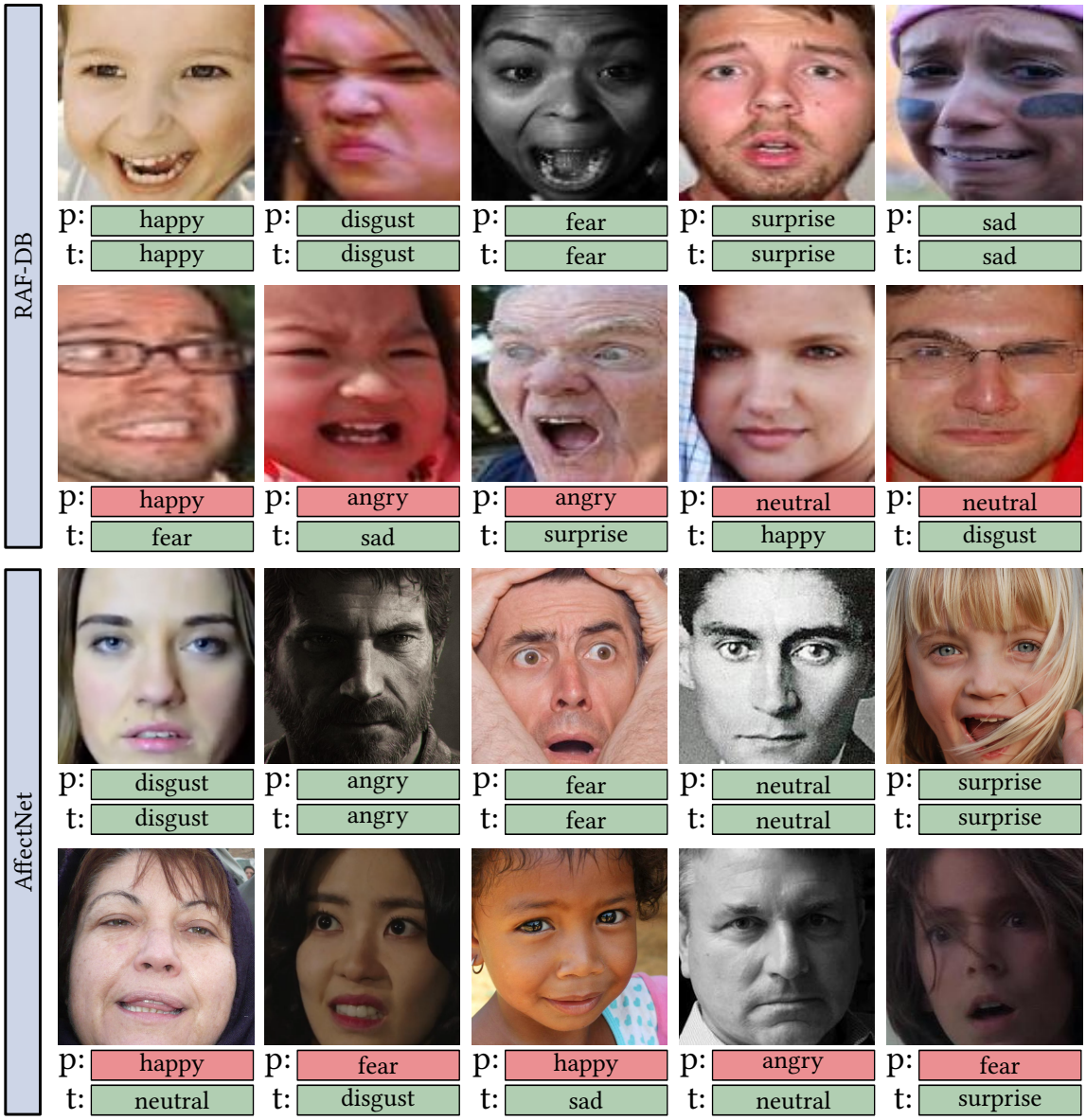


Fig. 4.4: Sample correctly classified and misclassified images from RAF-DB and AffectNet from the model trained with DACL method. "p" is for prediction and "t" is for true label.

the average accuracy on RAF-DB when compared to other methods.

Figure 4.5 presents the confusion matrices obtained by the best softmax loss method, the best center loss method, the best proposed DDA loss method, and the proposed DACL method on RAF-DB. Similarly, Figure 4.6 presents the confusion matrices obtained by the best softmax loss method, the best center loss method, the best proposed DDA loss method, and the proposed DACL method on AffectNet. All these matrices evaluate the recognition accuracy of individual classes. g-DACL is excluded since DACL achieves better performance.

RAF-DB. Center loss boosts the recognition accuracy of most classes except *disgust* and *surprise* compared to softmax loss. DDA loss improves on center loss by maintaining comparable results for most classes except *neutral* while boosting the recognition accuracy of the minority classes *fear* and *disgust*. As a result, we achieve the same trend of improvement as discussed in Chapter 3 over baseline methods. On the other hand, DACL boosts the recognition accuracy of all classes in RAF-DB's testing set except *happy* and *anger* compared to the DDA loss. The overall performance of DACL is better for three reasons: 1. It achieves significantly higher recognition accuracy for *surprise* compared to the center loss and DDA loss, 2. It achieves significantly higher recognition accuracy for *neutral* compared to softmax loss and DDA loss, and 3. It maintains comparable results for other classes compared to the other methods. It is noteworthy that DACL maintains comparable results on minority classes *anger* and *fear* while boosting *disgust* compared to DDA loss.

AffectNet. Center loss boosts the recognition accuracy of most classes except *disgust*. DDA loss improves on center loss by maintaining comparable results for most classes while boosting the recognition accuracy for *happy*. While DDA loss does not boost the recognition accuracy of minority classes in this case, the recognition accuracy boost for *happy* can be explained by the better separation of feature clusters for all classes in the embedding space, as mentioned in Chapter 3. On the other hand, DACL significantly boosts the recognition accuracy of *neutral*, *sad*, *fear*, and *disgust* while maintaining comparable results for the rest

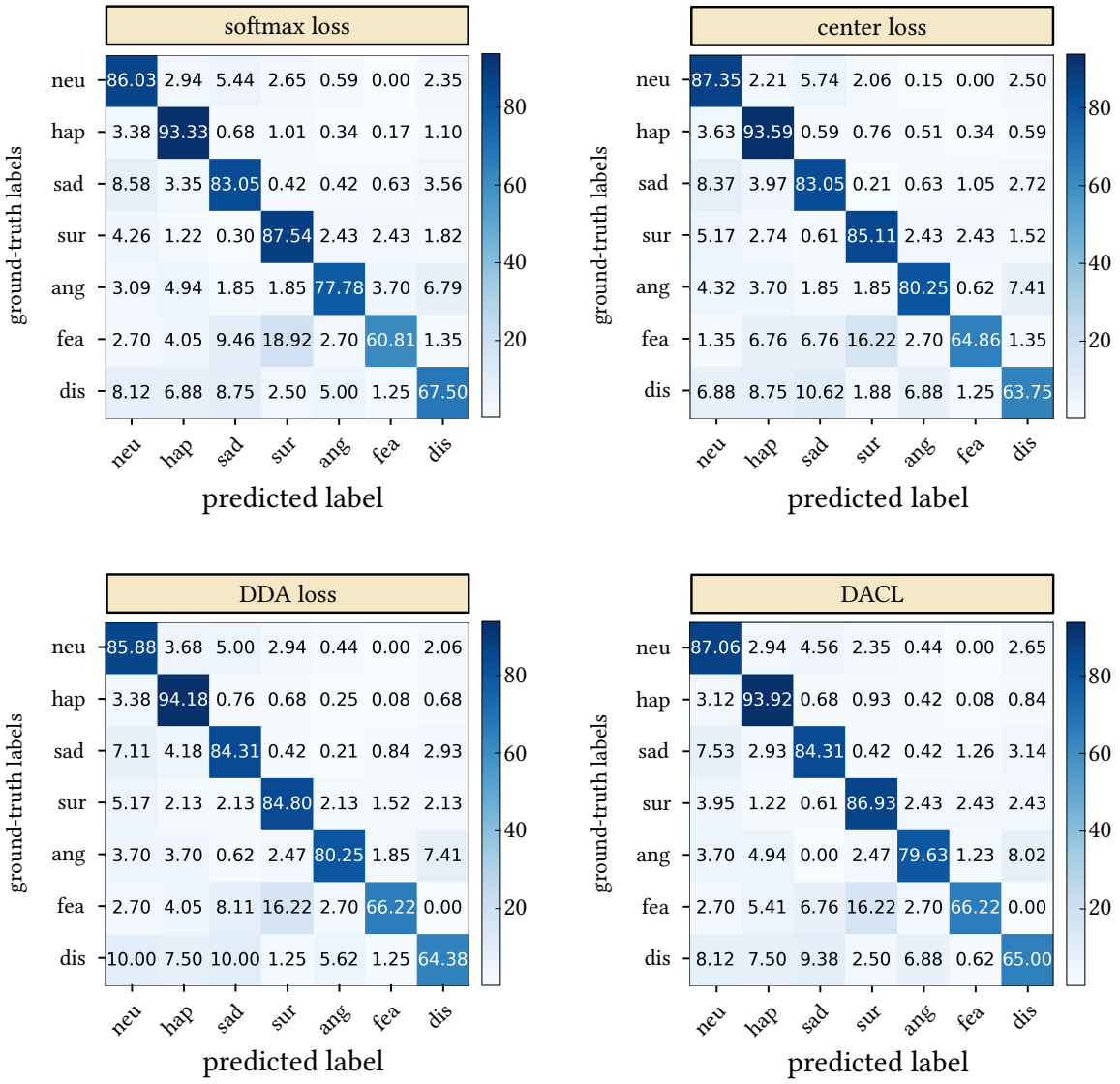


Fig. 4.5: Confusion matrices for the recognition accuracy of RAF-DB using baseline methods and the proposed method. All the models are pre-trained with *MS-CELEB-1M* dataset.

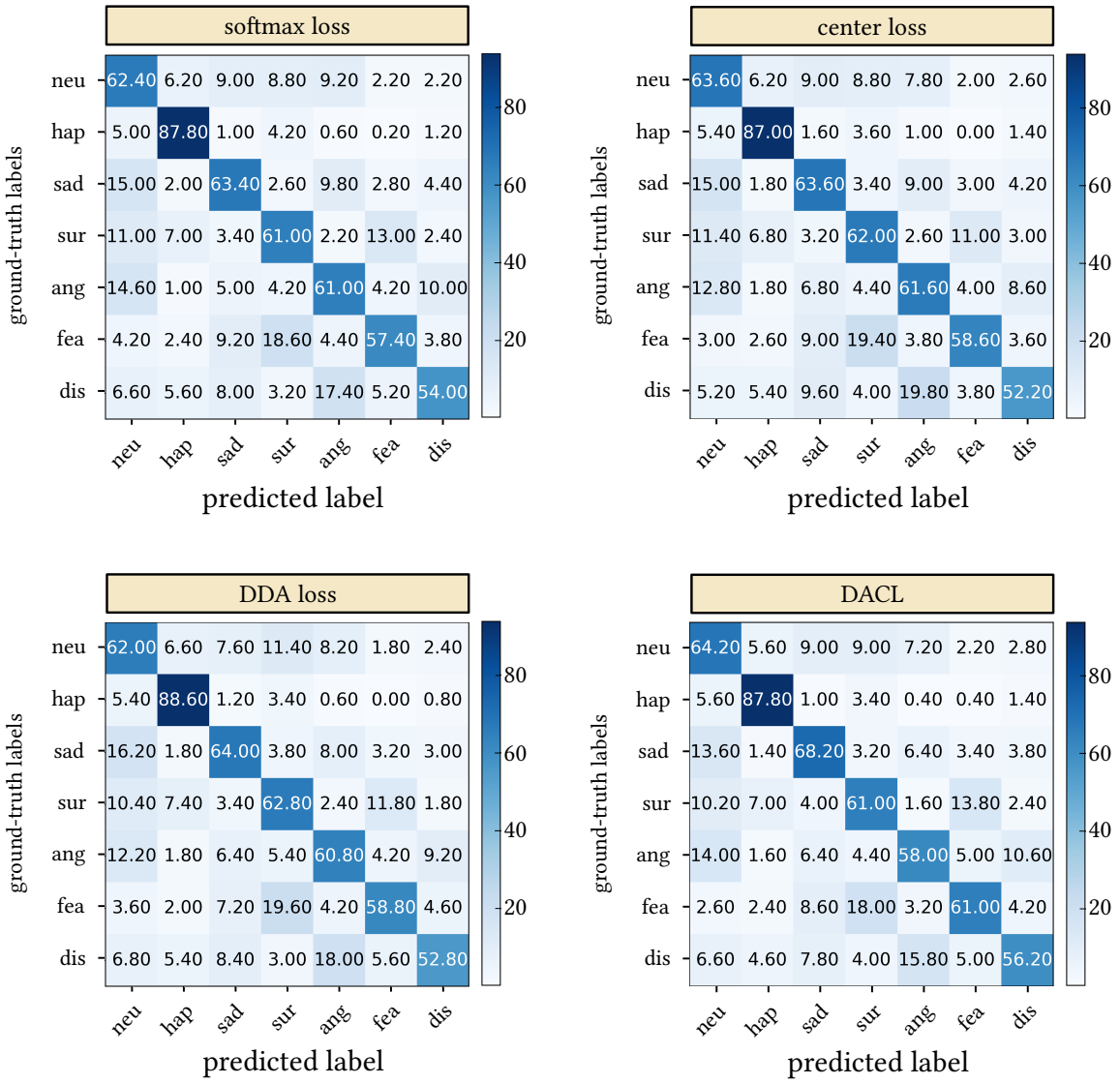


Fig. 4.6: Confusion matrices for the recognition accuracy of AffectNet using baseline methods and the proposed method. All the models are pre-trained with *MS-CELEB-1M* dataset.

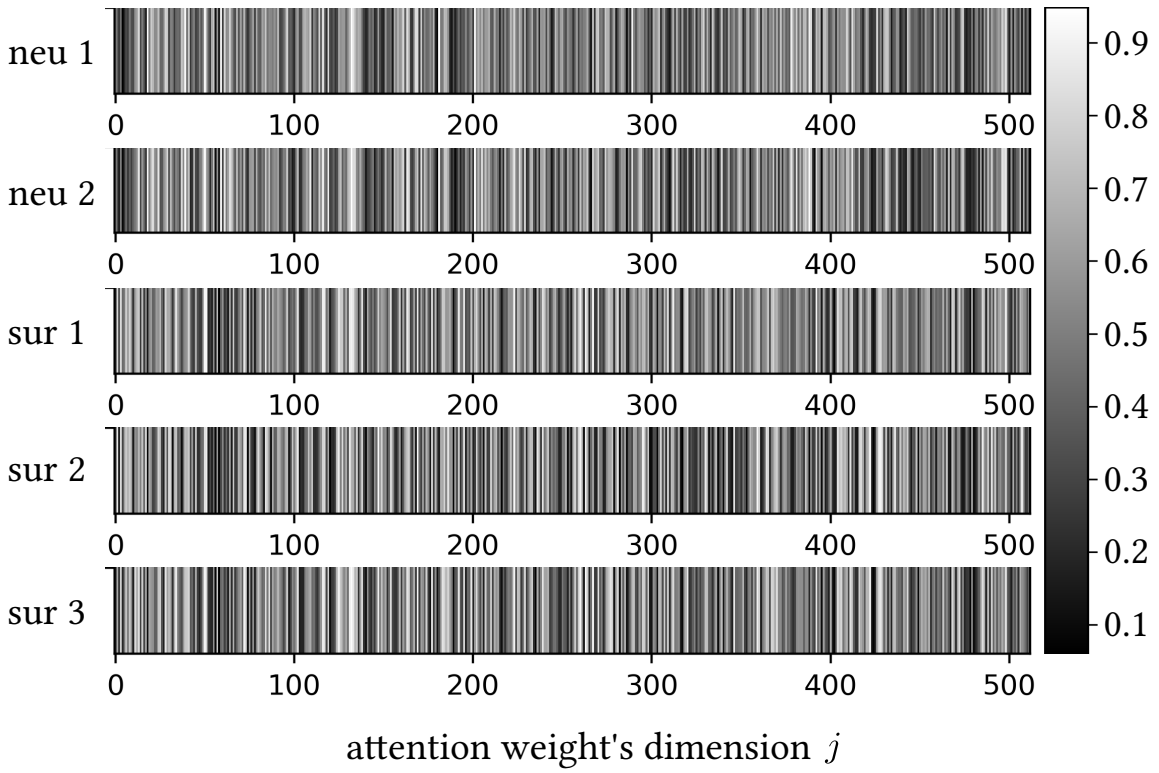


Fig. 4.7: Visualization of attention weights

of the classes. However, DACL performs poorly on *anger* when compared to other methods.

We proposed g-DACL as a variant of DACL primarily to demonstrate one example of how our hybrid method can be customized to match a specific task. Additionally, we hypothesized that by capturing more complex relationships, we could improve the recognition results. However, except for delivering comparable results on AffectNet, g-DACL performs slightly worse than DACL and DDA loss on RAF-DB. This is perhaps due to the following two reasons: 1. g-DACL captures more complex relationships than necessary from the context feature, and 2. g-DACL models the noise information in the dataset that lead to over-fitting.

4.3.4 Attention Weights Visualization

To demonstrate the interpretability of our proposed approach, we have illustrated the 512-dimensional attention weights in Figure 4.7. We have randomly selected two learned

attention weight vectors from the *neutral* class, and three learned attention weights from the *surprise* class. It is clear that the learned attention weights from the same classes follow very similar patterns, and the attention weights from different classes are not similar. For instance, both *neutral* samples exhibit attention weights that are filtered out around dimensions 0, 150, 190, 480, and 500. On the other hand, all samples from the *surprise* class depict attention weights that are filtered out around dimensions 50, 140, 220, and 480. Evidently, the *surprise* 2 and *surprise* 3 samples have learned almost identical attention weights. Consequently, we can verify that DACL adaptively learns the contribution of features along each dimension in the DML's objective function.

4.3.5 Conclusion

In this chapter, we propose a flexible method called Deep Attentive Center Loss (DACL) and its gated variant, g-DACL, for Facial Expression Recognition (FER) under wild scenarios. DACL and g-DACL are hybrid approaches that utilize a Deep Metric Learning (DML) method and an attention mechanism. Specifically, our hybrid approach takes advantage of a sparse re-formulation of center loss to adaptively control the contribution of the deep feature representations in the DML's objective function. Additionally, an attention mechanism that is fully parameterized by a customizable neural network estimates the probability of contribution along all dimensions by providing attention weights to the sparse center loss. We empirically show that DACL outperforms our baseline methods (softmax loss and center loss), g-DACL, and other state-of-the-art methods on two wild FER datasets, namely RAF-DB and AffectNet. DACL can be easily customized to match specific tasks other than FER. Moreover, the proposed approach is easily extensible with other DML objective functions.

CHAPTER 5

CONCLUSION

We have observed significant scientific advances in the field of Computer Vision over the past decade. Analyzing facial expressions is such active field of research in Computer Vision that has demonstrated significant progress. Facial Expression Recognition (FER) is an essential visual technology to detect emotion, given the input to the intelligence system is a facial image.

Due to the progress in deep learning research, Convolutional Neural Networks (CNNs) have demonstrated significant performance in visual recognition tasks. Notably, FER methods that are particularly based on CNNs have significantly outperformed conventional methods in FER. However, there are some obstacles when a FER model is developed for real-world applications. Real-world FER requires a massive corpus of annotated images acquired in an unconstrained environment, namely wild FER datasets. A large-scale wild dataset is comprised of images that are broadly varied in the pose, gender, age, illumination, demography, and image quality. Accordingly, there are two challenges associated with wild FER datasets: 1. Large intra-class variation and inter-class similarity, and 2. Extremely imbalanced distribution of data, an intrinsic issue in the wild dataset acquisition.

Conventional practices in deep learning yield prediction models with sub-optimal performance. In this dissertation, we focus on developing models and techniques inspired by Deep Metric Learning (DML) to tackle the the two aforementioned challenges with FER under wild scenarios and improve over state-of-the-art methods.

In Chapter 3, we introduce Discriminant Distribution-Agnostic loss (DDA loss) to enhance the discrimination power of widely used softmax loss. DDA loss regularizes the embedding space such that the deep features are well-clustered. DDA loss achieves intra-class compactness and inter-class separation by implicitly pushing the deep features of a class away from other classes and pulling them toward each other. Supervised jointly

by softmax loss and center loss, DDA loss efficiently segregates the deep features for both majority and minority classes. We show that DDA loss is trivially optimized by the standard Stochastic Gradient Descent (SGD) algorithm. Hence, it can be employed readily for any visual recognition task with any state-of-the-art CNN architecture. We visually analyze the behavior of our proposed DDA loss compared to the baseline methods (softmax loss and center loss) with a synthetically generated wild dataset. Extensive experiments on two widely popular wild FER datasets demonstrate the superior performance of our proposed DDA loss.

In Chapter 4, we approach the problem from a different perspective. We argue that the conventional DML approaches are prone to over-fitting and hence deliver sub-optimal generalization. Particularly, we develop a hybrid approach called Deep Attentive Center Loss (DACL) and its gated variant called g-DACL to tackle the FER in the wild. Both DACL and g-DACL methods utilize a sparse re-formulation of center loss, namely, the sparse center loss, to selectively discriminate a deep feature along its dimensions in the embedding space. This sparse center loss is jointly optimized with softmax loss and is trained using the standard SGD algorithm. We further design an extensible and modular attention mechanism that can be integrated in any CNN architectures to accommodate the sparse center loss with adaptive attention weights. The attention weights estimate the contribution of the deep feature across its dimensions based on the input. The sparse center loss and the attention mechanism comprise the building blocks of the proposed DACL method. Extensive experiments demonstrate the superior performance of DACL compared to state-of-the-art methods and the baseline methods (softmax loss and center loss) on two widely used wild FER datasets. We also observe that g-DACL performs similarly as the naive DACL. DACL can be easily modified and extended to other DML objective functions.

Our proposed methods mainly fall under the category of DML, where the underlying prediction model is capable of yielding highly discriminative features. All models are trained in an end-to-end learning framework with many practical advantages: 1. Every module in the model is built on top of the previous modules, offering less code complexity, 2. It

allows the modules to be trained in parallel, taking the advantage of GPUs' concurrent computations, and 3. Prediction models can be easily customized to match a specific task.

One advantage of DACL over DDA loss is that it does not need to tune any hyper-parameters, which makes training easier. Another advantage of DACL is that it focuses on improving the recognition accuracy of all classes, while DDA loss focuses more on the minority classes. However, the integrated module in DACL adds to the computation overhead during training. As a result, a trade-off has to be considered when choosing the best method for a specic application.

We also acknowledge that there are remaining challenges associated with FER under wild scenarios. We briefly review a few of them to inspire future research:

- Accurately annotating FER datasets is an impossible task. When a research group is assigned to annotate a large-scale dataset, label errors are inevitable. Although a small amount of noise might improve generalization, training datasets that exhibit large label noise are detrimental to the learning algorithm.

- Facial expressions are subjectively interpreted when annotated. Most wild FER datasets acquired from the web are annotated using crowd-sourcing techniques where human annotators are hired to look at data and tag them. One annotator might have a different interpretation of happiness by observing a smile, while another might annotate the same image with neutral. Consequently, a bias is developed toward labeling large-scale datasets.

- Choosing proper pre-training weights has been a major challenge in our experiments. As explored in Chapter 3, *ImageNet* does not always improve the recognition results. However, our experiments in Chapter 4 with models pre-trained with *MS-CELEB-1M* dataset proved otherwise. More investigation on this paradigm may lead to better performance for FER applications.

Future research should target the challenges mentioned above as they are central to research in FER. Additionally, visual contexts such as background, objects in the scene,

gaze, positional relationship with other subjects might reveal a more accurate emotion than merely facial image [73]. These different visual cues surrounding the subject's face (*i.e.,* context) can be be interpreted and incorporated for emotion recognition.

In this dissertation, we exclusively explored DACL with center loss. However, DACL is a modular framework that can be applied to any DML approach. Hence, DACL integrated with DDA loss might improve the recognition accuracies while enjoying the benefits of well-separated feature clusters in the embedding space. We also encourage future researchers to explore different architecture for the attention network in DACL. One possible variation is a convolutional Context Encoder Unit (CE-Unit) to capture spatial relationships between the encoder layers.

Other possible improvements include but not limited to optimizing hyper-parameters, utilizing Recurrent Neural Networks (RNNs) to estimate the attention weights in DACL, employing different clustering methods in the embedding space (e.g., Non-negative Matrix Factorization), and utilizing ensemble methods.

REFERENCES

[1] P. Ekman and W. V. Friesen, *Facial Action Coding System.* Consulting Psychologists Press, 1978.

[2] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[3] E. Schubert, "Measuring emotion continuously: Validity and reliability of the two-dimensional emotion-space," *Australian Journal of Psychology*, vol. 51, no. 3, pp. 154–165, 1999.

[4] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124–129, 1971.

[5] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[6] G. Zhao and M. Pietikainen, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[7] R. Zhi, M. Flierl, Q. Ruan, and W. B. Kleijn, "Graph-Preserving Sparse Nonnegative Matrix Factorization With Application to Facial Expression Recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 38–52, 2011.

[8] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, "DISFA: A Spontaneous Facial Action Intensity Database," *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 151–160, 2013.

[9] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2562–2569.

[10] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in Representation Learning: A Report on Three Machine Learning Contests," in *Neural Information Processing*, 2013, pp. 117–124.

[11] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon, "Video and Image based Emotion Recognition Challenges in the Wild: EmotiW 2015," in *ACM on International Conference on Multimodal Interaction*, 2015, pp. 423–426.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[19] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning*, 2013, pp. 1139–1147.

[20] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems 4*, 1992, pp. 950–957.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8026–8037.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010, pp. 94–101.

[24] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *IEEE International Conference on Multimedia and Expo*, 2005, pp. 5 pp.–.

[25] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 200–205.

[26] M. Taini, G. Zhao, S. Z. Li, and M. Pietikainen, "Facial expression recognition from near-infrared video sequences," in *International Conference on Pattern Recognition*, 2008, pp. 1–4.

[27] S. Li, W. Deng, and J. Du, "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2584–2593.

[28] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2019.

[29] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the Loss Layer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4753–4762.

[30] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[31] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[32] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 1735–1742.

[33] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[34] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-Aware Convolutional Neural Network for Facial Expression Recognition," in *IEEE International Conference on Automatic Face Gesture Recognition*, 2017, pp. 558–565.

[35] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 2106–2112.

[36] Y. Guo, D. Tao, J. Yu, H. Xiong, Y. Li, and D. Tao, "Deep Neural Networks with Relativity Learning for facial expression recognition," in *IEEE International Conference on Multimedia Expo Workshos*, 2016, pp. 1–6.

[37] X. Liu, B. V. K. V. Kumar, J. You, and P. Jia, "Adaptive Deep Metric Learning for Identity-Aware Facial Expression Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 522–531.

[38] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, pp. 1857–1865.

[39] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep Relative Distance Learning: Tell the Difference between Similar Vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2167–2175.

[40] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 499–515.

[41] S. Li and W. Deng, "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2019.

[42] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island Loss for Learning Discriminative Features in Facial Expression Recognition," in *IEEE International Conference on Automatic Face Gesture Recognition*, 2018, pp. 302–309.

[43] Y. Li, Y. Lu, J. Li, and G. Lu, "Separate loss for basic and compound facial expression recognition in the wild," in *Asian Conference on Machine Learning (ACML)*, vol. 101, 2019, pp. 897–911.

[44] Z. Li, S. Wu, and G. Xiao, "Facial Expression Recognition by Multi-Scale CNN with Regularized Center Loss," in *International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3384–3389.

[45] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face Alignment at 3000 FPS via Regressing Local Binary Features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1685–1692.

[46] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 2035–2043.

[47] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[48] Y. Li, J. Zeng, S. Shan, and X. Chen, "Patch-Gated CNN for Occlusion-aware Facial Expression Recognition," in *International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2209–2214.

[49] J. Zhang, M. Kan, S. Shan, and X. Chen, "Occlusion-Free Face Alignment: Deep Regression Networks Coupled with De-Corrupt AutoEncoders," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3428–3437.

[50] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2439–2450, 2019.

[51] S. Zhao, H. Cai, H. Liu, J. Zhang, and S. Chen, "Feature selection mechanism in CNNs for facial expression recognition," in *British Machine Vision Conference (BMVC)*, 2018.

[52] C. Florea, L. Florea, M. A. Badea, and C. Vertan, "Annealed label transfer for face expression recognition," in *British Machine Vision Conference (BMVC)*, 2019.

[53] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *ACM International Conference on Multimodal Interaction (ICMI)*, 2016, pp. 279–283.

[54] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1867–1874.

[55] J. Zeng, S. Shan, and X. Chen, "Facial Expression Recognition with Inconsistently Annotated Datasets," in *European Conference on Computer Vision (ECCV)*, Cham, 2018, pp. 227–243.

[56] J. Lee, S. Kim, S. Kim, J. Park, and K. Sohn, "Context-Aware Emotion Recognition Networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 10 143–10 152.

[57] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Acted facial expressions in the wild database," Tech. Rep., 2011.

[58] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[59] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2016, pp. 507–516.

[60] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6738–6746.

[61] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking Feature Distribution for Loss Functions in Image Classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9117–9126.

[62] M. Hayat, S. Khan, S. W. Zamir, J. Shen, and L. Shao, "Gaussian Affinity for Max-Margin Class Imbalanced Learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6468–6478.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[64] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5353–5360.

[65] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[67] K. He, R. Girshick, and P. Dollar, "Rethinking ImageNet Pre-Training," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4917–4926.

[68] S. Li and W. Deng, "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2019.

[69] D. Kollias and S. Zafeiriou, "Expression, affect, action unit recognition: Aff-wild2, multi-task learning and ArcFace," in *British Machine Vision Conference (BMVC)*, 2019, p. 297.

[70] C. Wang, S. Wang, and G. Liang, "Identity- and pose-robust facial expression recognition through adversarial feature learning," in *ACM International Conference on Multimedia*, 2019, pp. 238–246.

[71] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International Conference on Machine Learning (ICML)*, 2017, pp. 933–941.

[72] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision (ECCV)*, 2016.

[73] A. M. Martinez, "Context may reveal how you feel," *Proceedings of the National Academy of Sciences*, vol. 116, no. 15, p. 7169, 2019.

CURRICULUM VITAE

Amir Hossein Farzaneh

Education

- Ph.D., Computer Science, Utah State University, Logan, Utah, US, Adviser: Xiaojun Qi, June 2020.
- M.Sc., Electrical Engineering, Shahrood University of Technology, Shahrood, Iran, 2015.
- B.Sc., Electrical Engineering, Shahid Beheshti University, Tehran, Iran, 2012.

Research Interests

- Computer Vision
- Deep Learning
- Metric Learning
- Machine Learning

Published Journal Articles

- Amir Hossein Farzaneh and Xiaojun Qi, "Cross-spectral registration of natural images with SIPCFE", *Machine Vision and Applications*, Vol. 31 (1), Pages 10, 2020.
- Mohammadreza Javanmardi, Amir Hossein Farzaneh and Xiaojun Qi, "A Robust Structured Tracker Using Local Deep Features", *Electronics*, Vol. 9 (5), Pages 846, 2020.

Published Conference Papers

- Amir Hossein Farzaneh and Xiaojun Qi, "Facial Expression Recognition in the Wild via Deep Attentive Center Loss", *in ACM International Conference on Multimedia (ACM MM)*, Seattle, WA, US, October 2020 (**Under Review**).

- Amir Hossein Farzaneh and Xiaojun Qi, "Discriminant Distribution-Agnostic Loss for Facial Expression Recognition in the Wild", *in IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, US, June 2020.

- Amir Hossein Farzaneh, Yanghee Kim, Mengxi Zhou, and Xiaojun Qi, "Developing a Deep Learning-Based Affect Recognition System for Young Children", *in International Conference on Artificial Intelligence in Education (AIED)*, Chicago, IL, US, June 2019.

- Amir Hossein Farzaneh and Xiaojun Qi, "Optimized Feature-Based Image Registration for RGB and NIR Pairs", *in IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, US, July 2018.